Technology
Arts Sciences
TH Köln

Fachhochschule
Dortmund
University of Applied Sciences and Arts

Verbundstudiengang Wirtschaftsinformatik

Abschlussarbeit

zur Erlangung

des Bachelorgrades

Bachelor of Science

in der Fachrichtung Informatik

“Vergleich und Bewertung der Datenbanksicherheit von relationalen Datenbanksystemen“

Erstprüferin:	Prof. Dr. Birgit Bertelsmeier
Zweitprüfer:	M.Sc. Andre Kasper
vorgelegt am:	15. Juli 2016
von cand.	Markus Berg
Email:	Markus.Berg@smail.th-koeln.de
Matr.-Nr.:	011 070 545

Zusammenfassung

Der digitale Schwarzmarkt mit gestohlenen Daten floriert. Alleine im Jahr 2015 wurden über eine halbe Milliarde Datensätze aus Datenbanksystemen entwendet. Viele Unternehmen bemerken den Sicherheitsvorfall selbst nicht oder geben aus Angst vor einem Imageschaden den Vorfall nicht bekannt. Sind die Datenbestände nicht durch effektive Sicherheitsmechanismen vor Datendiebstahl geschützt, können Cyberkriminelle leicht aus den gestohlenen Daten Kapital schlagen. Die Softwarehersteller von Datenbanksystemen bieten für ihre Produkte unterschiedliche Schutzvorkehrungen an. Ziel dieser Ausarbeitung ist es, die Bedrohungen und die Sicherheitsrisiken von Datenbanksystemen aufzuzeigen und daraus die Aspekte der Datenbanksicherheit abzuleiten. Die ausgearbeiteten Sicherheitsaspekte bilden die Grundlage für den Vergleich der Datenbanksicherheit von relationalen Datenbanklösungen. Die anschließende Bewertung dient dem Nachweis und der Überprüfbarkeit der Datenbanksicherheit relationaler Datenbanksysteme.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

2

2FA ..... *Two-Factor-Authentication*

3

3DES..... *Triple Data Encryption Standard*

A

AAA..... *Authentifizierung, Autorisierung, Abrechnung*
AES..... *Advanced Encryption Standard*
AGPL ..... *Affero General Public License*
API ..... *Application Programming Interface*
App..... *Application Software*

B

BDSG..... *Bundesdatenschutzgesetz*
BSD-License ..... *Berkeley Software Distribution License*
BSI ..... *Bundesamt für Sicherheit in der Informationstechnik*

C

CBAC ..... *Code-Based Access Control*
CBC ..... *Cipher Block Chaining*
CMS ..... *Content Management System*
CPU ..... *Central Processing Unit*
CVE..... *Common Vulnerabilities and Exposures*
CVSS ..... *Common Vulnerability Scoring System*

D

DAC ..... *Discretionary Access Control*
DB ..... *Datenbank*
DBMS..... *Datenbankmanagementsystem*
DBS..... *Datenbanksystem*
DBUA ..... *Database Upgrade Assistant*
DCL..... *Data Control Language*
DDL..... *Data Definition Language*
DES..... *Data Encryption Standard*
DML ..... *Data Manipulation Language*
DV ..... *Database Vault*

E

ECB..... *Electronic Code Book*

G

GSSAPI..... *Generic Security Service Application Program Interface*

H

HTTP..... *Hypertext Transfer Protocol*

I

IBM..... *International Business Machines Corporation*
IP ..... *Internet Protocol*
IT ..... *Informationstechnik*
IT-System..... *Informationstechnisches System*

L

LBAC..... *Label-Based Access Control*
LDAP..... *Lightweight Directory Access Protocol*

M

MAC ..... *Mandatory Access Control*
MD5 ..... *Message-Digest Algorithm 5*
MIT-License ..... *Massachusetts Institute of Technology License*

N

NIST..... *National Institute of Standards and Technology*
NoSQL ..... *Not only SQL*
NSA..... *National Security Agency*
NVD ..... *National Vulnerability Database*

O

OLS..... *Oracle Label Security*
OUI..... *Oracle Universal Installer*

P

PAM ..... *Pluggable Authentication Module*
PAP..... *Password Authentication Protocol*
PBKDF2 ..... *Password-Based Key Derivation Function 2*
PC ..... *Personal Computer*
PGDG ..... *PostgreSQL Global Development Group*
PGP ..... *Pretty Good Privacy*
PKCS12 ..... *Public-Key Cryptography Standards 12*
PKI ..... *Public-Key-Infrastructure*
PL/SQL ..... *Procedural Language/Structured Query Language*

R

RADIUS ..... *Remote Authentication Dial-In User Service*
RAM ..... *Random-Access Memory*
RAS ..... *Real Application Security*
RAW ..... *Rohdaten*
RBAC ..... *Role-Based Access Control*
RC4 ..... *Ron's Code 4*
RSA ..... *Rivest, Shamir und Adleman*
RSI ..... *Relational Software Inc.*

S

SASL ..... *Simple Authentication and Security Layer*
SDL ..... *Software Development Laboratories*
SELinux ..... *Security Enhanced Linux*
Sepgsql ..... *Security Enhanced PostgreSQL*
SHA ..... *Secure Hash Algorithm*
SMS ..... *Short Message Service*
SQL ..... *Structured Query Language*
SSL ..... *Secure Sockets Layer*
SSPI ..... *Security Support Provider Interface*

T

TCP/IP ..... *Transmission Control Protocol/Internet Protocol*
TDE ..... *Transparent Data Encryption*
TGS ..... *Ticket Granting Service*
TGT ..... *Ticket Granting Ticket*
TLS ..... *Transport Layer Security*

V

VPD ..... *Virtual Private Database*

X

XML ..... *Extensible Markup Language*
XOR ..... *Exklusiv-Oder*

1 Einleitung

Datendiebstahl, Datenpannen sowie digitale Betrugsversuche sind immer häufiger Bestandteil von Schlagzeilen. Aktuelles Negativbeispiel ist das Geschäftskontakt- und Karriere-Netzwerk LinkedIn, das im Juni 2012 von Cyberkriminellen gehackt wurde. Nach dem Datenklau wurden sechs Millionen Passwörter (ohne zugehörige E-Mail-Adressen) von Benutzern auf einer russischen Webseite veröffentlicht.[1] Das Unternehmen kommentierte nie den Vorfall und seine Ausmaße und sprach lediglich von einigen betroffenen Benutzern. Vier Jahre später, im Mai 2016 kam die ganze Wahrheit der Datenpanne ans Licht: Unbekannte boten die komplette Nutzerdatenbank aus dem Datenraub von 2012 im Darknet zum Verkauf an.[2] LinkedIn veröffentlichte daraufhin am 18. Mai 2016 die Pressemeldung, dass über 100 Millionen Benutzerdaten (E-Mail-Adressen und Passwörter) aus der Datenpanne von 2012 aufgetaucht sind.[3] Die Passwörter waren nur mit dem schwachen Secure Hash Algorithm 1 ohne „Salt“ (in der Kryptologie eine zufällig erzeugte Zeichenkette, die an das Passwort zur Stärkung angehängt wird) verschlüsselt. Der größte Teil der Benutzerpasswörter war bereits wenige Tage nach der Veröffentlichung entschlüsselt und wurde für weitere kriminelle Machenschaften (z.B. Trojaner-Mails) verwendet. Bei der Nachbetrachtung stellte sich auch heraus, dass “123456“ das beliebteste Passwort bei den LinkedIn-Benutzern war; es wurde über eine Million Mal verwendet.[4]

1.1 Motivation

Eine Datenpanne liegt vor, wenn Dritte an sensible personenbezogene Daten gelangen. Laut der Ponemon-Studie „2016 Cost of Data Breach Study: Global Analysis“ sind 52 Prozent der deutschen Datenpannen durch Angriffe von Cyberkriminellen hervorgerufen worden. Weitere 30 Prozent bedingt durch Systemfehler und die restlichen 18 Prozent sind auf Benutzerfehler zurückzuführen. Im Durchschnitt verursacht eine Datenpanne in Deutschland Schäden in Höhe von 3,61 Millionen Euro. Je länger ein Unternehmen zur Aufdeckung der Datenpanne benötigt, desto höher sind die Folgekosten.[5]

Die digitalen Angriffe auf Unternehmen, Behörden oder Privatleute sind im Vergleich zu den Vorjahren extrem angestiegen. Laut der BITKOM Studie „Digitale Angriffe auf jedes zweite Unternehmen“ aus dem Jahre 2015 sind 51 Prozent aller deutschen Unternehmen in den letzten zwei Jahren der Wirtschaftsspionage,

[1] Vgl. **[Clu12]**
[2] Vgl. **[Sch1]**
[3] Vgl. **[Cor16]**
[4] Vgl. **[SchJ]**
[5] Vgl. **[Pon16]** S. 11-12

Sabotage oder Datendiebstahl zum Opfer gefallen.[6] 2015 wurden über eine halbe Milliarde Datensätze mit personenbezogener Daten gestohlen.[7] Laut der Symantec Studie „Internet Security Threat Report“ aus dem Jahr 2016 hat sich die Anzahl der entdeckten Sicherheitslücken in Softwareprodukten im Vergleich zum Vorjahr mehr als verdoppelt.[8]

Die Studie „SAP Angriffsvektoren“ des SAP Sicherheitsspezialisten Onapsis aus dem Jahr 2015 legt offen, dass mehr als 95 Prozent der untersuchten Datenbanksysteme Schwachstellen aufweisen. Weitere Schwachstellen gibt es laut der Studie auch im Bereich der Kunden- und Lieferantenportale, die entweder mit zu niedrigen Sicherheitseinstellungen konfiguriert sind oder die Möglichkeit zum Einschleusen von Schadcode über proprietäre SAP-Protokolle nicht unterbinden. Cyberkriminelle nutzen diese Schwachstellen, um sensible Daten wie z.B. Kunden-, Kredit- und Finanzdaten abzugreifen. Darüber hinaus deckt die Studie auf, dass bei den meisten Unternehmen Sicherheitsaktualisierungen (Sicherheitsupdates) in den Datenbanksystemen erst nach 18 Monate oder später installiert werden. SAP veröffentlichte im Jahr 2014 für seine Produkte 391 Sicherheitsupdates, von denen ca. 50 Prozent als kritisch eingestuft wurden.[9]

Ein ungepatchtes Datenbanksystem ist leichte Beute für Cyberkriminelle, die mit einer Armee von Rechnern (Botnet) nach verwundbaren oder schwach konfigurierten Server-Plattformen suchen. Dabei ist die Zeitspanne zwischen dem Auftreten und dem Ausbessern des Softwarefehlers (Bug) entscheidend. Schwachstellen werden unmittelbar nach dem Bekanntwerden der Sicherheitslücke durch Hacker ausgenutzt. Werden die Datenbanksysteme nicht schnell und ausreichend geschützt, kann es zum Datenverlust oder zur Datenverfälschung kommen.

Die Sicherheit von Datenbanksystemen basiert auf einem komplexen Gebilde aus Verfahrensanweisungen (Policy), festgelegten Maßnahmen und technischen Faktoren. Die Datenbanksicherheit (Database Security) beschreibt eine Sammlung von Sicherheitsmechanismen zur Absicherung und zum Schutz der Datenbanksoftware vor unrechtmäßiger Nutzung, bösartigen Bedrohungen und Angriffen. Die einzelnen Sicherheitsmaßnahmen sind an die Informationssicherheit angelehnt, um die Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit von unternehmenskritischen Daten sicherzustellen.[10]

[6] Vgl. **[BITKOM]**
[7] Vgl. **[Sym16]** S. 6
[8] Vgl. ebd. S. 5
[9] Vgl. **[SAP15]**
[10] Vgl. **[Cas95]** S. IX

1.2 Zielsetzung

Ziel der Ausarbeitung ist es, die Datenbanksicherheit von relationalen Datenbanksystemen zu vergleichen und zu bewerten. Dazu werden die Sicherheitsrisiken von Datenbanksystemen näher betrachtet und die externen Bedrohungen kategorisiert. Anhand der benannten Sicherheitsrisiken und Bedrohungen werden sicherheitsrelevante Aspekte der Datenbanksicherheit abgeleitet. Diese bilden den Ausgangspunkt für den Vergleich und die Bewertung der Datenbanksicherheit relationaler Datenbanksysteme. Die Bewertung der Datenbanksicherheit erfolgt auf Basis der verankerten Kontroll- und Schutzmaßnahmen in den Datenbanksystemen MySQL, MariaDB, PostgreSQL und Oracle Database.

1.3 Vorgehen

Kapitel 2 fasst die theoretischen Grundlagen des behandelnden Themas zusammen. Hierfür wird die allgemeine Funktion sowie die interne Struktur eines Datenbanksystems näher betrachtet. Weiter wird erläutert, was unter Informationssicherheit bei Datenbanken zu verstehen ist. Zusätzlich wird aufgezeigt, welche Gesetze und Sicherheitsstandards speziell zum Datenschutz für Datenbanksysteme existieren. Kapitel 3 behandelt die Sicherheitsrisiken und die Bedrohungen von Datenbanksystemen. In Kapitel 4 erfolgt die Darstellung der einzelnen Sicherheitsmechanismen, die die Basis für die Bewertung bilden. Kapitel 5 stellt die unterschiedlichen relationalen Datenbanksysteme und die verankerten Schutzmaßnahmen vor. In Kapitel 6 werden die Datenbanksysteme MySQL, MariaDB, PostgreSQL und Oracle Database auf Grundlage der zuvor erarbeiteten Sicherheitsaspekte im Datenbankumfeld gegenübergestellt und bewertet. Am Ende werden die gewonnenen Erkenntnisse analysiert und zusammengefasst.

2 Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen für das Verständnis der weiteren Ausführungen gelegt. Hierzu werden die Themengebiete Datenbanksysteme und Informationssicherheit vertieft. Zunächst wird der Aufbau und die Architektur eines Datenbanksystems beschrieben. Danach werden die Schutzziele, die sich aus der Vertraulichkeit, der Integrität und der Verfügbarkeit ergeben, aufgeführt. Anschließend werden die technischen und organisatorischen Maßnahmen gemäß § 9 des Bundesdatenschutzgesetzes vorgestellt und auf die gesetzlichen Grundlagen von Datenbanksystemen eingegangen. Abschließend wird dargestellt, wie durch die Einhaltung von Sicherheitsstandards und Normen die Datenbanksicherheit optimiert werden kann.

2.1 Datenbanksysteme

Ein Datenbanksystem ist ein elektronisches System, das zur Speicherung, Wiedergewinnung, Verknüpfung und Auswertung sowie zur effizienten und widerspruchsfreien Aufbewahrung von digitalen Daten eingesetzt wird.[11] Das Datenbanksystem (DBS) besteht im Wesentlichen aus der Datenbank (DB) und dem Datenbankmanagementsystem (DBMS). Während die Datenbank zur elektronischen Datenverwaltung genutzt wird, dient das Datenbankmanagementsystem der Konfigurations- und Betriebsverwaltung der Datenbank. Darüber hinaus verfügt jedes Datenbanksystem über eine Kommunikationsschnittstelle, die zur Interaktion zwischen Anwendung und Benutzern sowie als Programmierschnittstelle (API) fungiert.[12] Die Art und Weise der Datenspeicherung wird durch das spezifische Datenmodell festgelegt. Grundsätzlich wird zwischen relationalen, objektorientierten und dokumentenbasierten Datenmodellen unterschieden. Die Ausarbeitung befasst sich ausschließlich mit relationalen Datenbanken. Hier werden die Daten in Tabellen gespeichert und durch Beziehungen (Relation) zueinander gestellt. Die Datenbanksprache zur Abfrage und Bearbeitung (Einfügen, Löschen und Verändern) der gehaltenen Daten ist bei relationalen Datenbanksystemen die sogenannte Structured Query Language (SQL). Datenbanksysteme, die nicht nur den relationalen Ansatz verfolgen, werden auch als „not only SQL“ (NoSQL) bezeichnet.[13]

Die interne Struktur eines Datenbanksystems ist in Abbildung 1 dargestellt. Anhand dieser Aufteilung lassen sich die wesentlichen Teilaspekte der Datenbanksicherheit von relationalen Datenbanksystemen erläutern.

[11] Vgl. **[Eng93]** S. 317
[12] Vgl. **[Cor11]** Kap. 16.3, S. 331
[13] Vgl. ebd. Kap. 2.3, S. 13-15

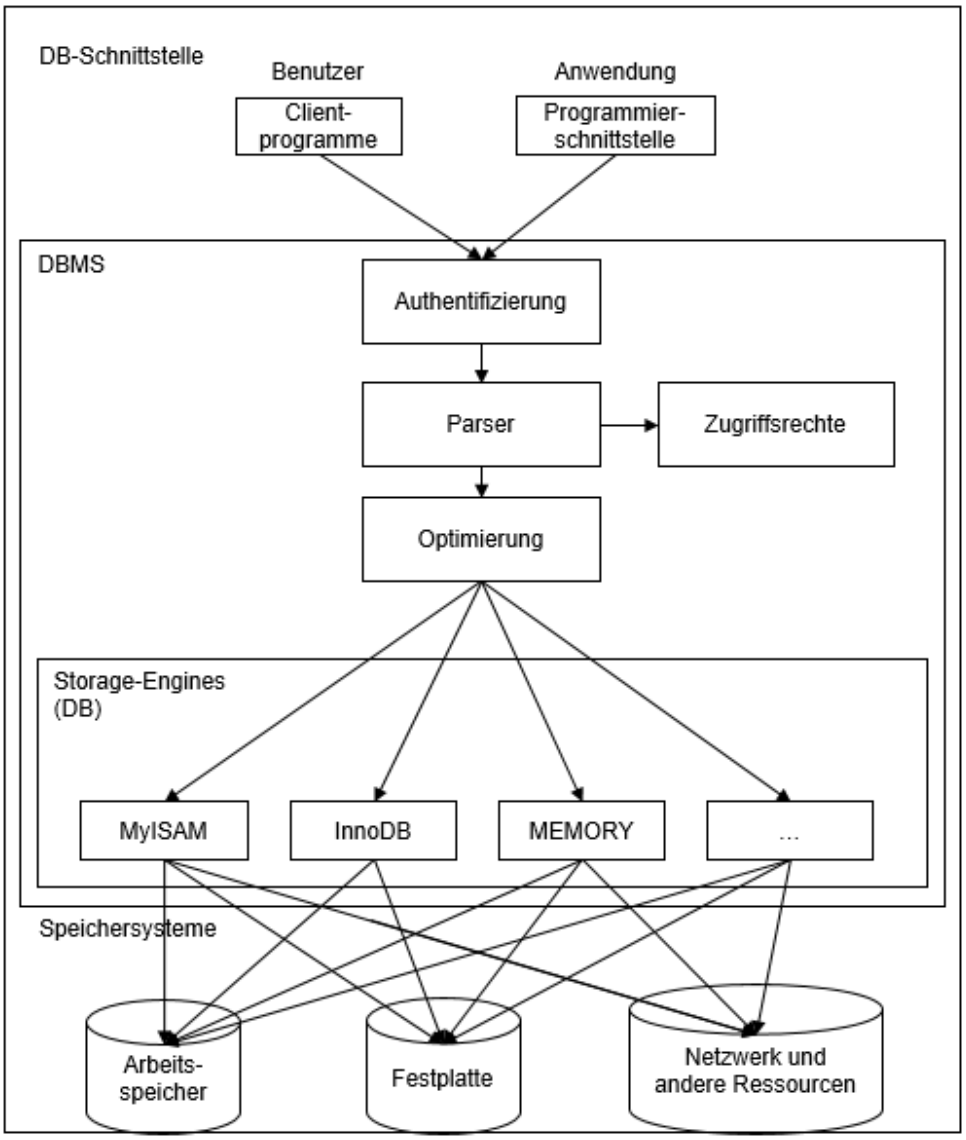

Abbildung 1 - Architektur eines Datenbanksystems[14]

Für die Kommunikation mit einem Datenbanksystem stehen dem Benutzer unterschiedliche Clientprogramme (z.B. MYSQL Client, Heidi SQL oder SQL*Plus) zur Verfügung. Anwendungen (z.B. Webanwendung eines Online-Shops) nutzen entsprechende Programmierschnittstellen (z.B. JDBC). Die Kommunikation wird mit dem Netzwerkprotokoll Transmission Control Protocol/Internet Protocol (TCP/IP) geregelt.[15] Das Datenbanksystem wird vor unbefugten Zugriff durch die Authentifizierung (Authentication) geschützt. Das bekannteste Authentifizierungsverfahren ist die Benutzername-Passwort-Authentifizierung. Neben dieser Authentifizierungsmethode existieren weitere Möglichkeiten (z.B. die Public-Key-Infrastructure), um den Zugang zu Datenbanksystemen effektiv zu schützen. Jede Datenbankanfrage wird durch den Parser auf syntaktische

[14] Angelehnt an **[Raz14]** Kap. 1
[15] Vgl. **[Mei12]** Kap. 2.1.2, S. 39-40

Korrektheit verifiziert. Danach wird überprüft, ob der zuvor autorisierte Benutzer auch über die Zugriffsrechte (Permission) für die Ausführung der Anfrage verfügt. Im nächsten Schritt wird die SQL-Anfrage zerlegt und nach effizienten Ausführungsmöglichkeiten zur Performancesteigerung untersucht. Diese Prozedur erfolgt im Hintergrund und kann durch den Benutzer nicht beeinflusst werden. In den Speichersubsystemen (Storage-Engine) werden die Daten physisch gespeichert. Die unterschiedlichen Speichersubsysteme (z.B. MyISAM, InnoDB) unterscheiden sich im Funktionsumfang, in der Abfragegeschwindigkeit und in der maximalen Speichergröße. Die My Indexed Sequential Access Method (MyISAM) ist ein internes Speichersubsystem des Datenbanksystems MySQL. Das Speichersubsystem InnoDB ist ein freies, alternatives Speichersubsystem, das von vielen Datenbanksystemen unterstützt wird und bevorzugt zum Einsatz kommt. Das MEMORY Speichersubsystem wird nur im Arbeitsspeicher (In-Memory) abgebildet.[16] Die Speichersubsysteme nutzen die Hardwareressourcen wie den Arbeitsspeicher (RAM), den Prozessor (CPU) oder die Netzwerkschnittstelle der IT-Infrastruktur des Datenbankservers.[17]

2.2 Informationssicherheit

Sensible Informationen wie Unternehmensdaten oder personenbezogene Daten unterliegen einem hohen Schutzbedarf. Informationen werden in der Informationstechnik (IT) als Daten verstanden, die spezielle Attribute aufweisen.[18] Der Schutz der zu verarbeitenden Informationen (Informationssicherheit) wird durch die verankerten Schutzziele, die sich aus der Vertraulichkeit von Informationen, der Integrität von Daten und der Verfügbarkeit von IT-Systemen zusammensetzen, sichergestellt. Mit der Informationssicherheit sollen Bedrohungen und Gefahren (siehe Kapitel 3) minimiert und der Verlust der damit verbundenen Schutzziele (Vertraulichkeit, Integrität und Verfügbarkeit) verhindert werden.[19]

Der Schutz der Vertraulichkeit verfolgt das Ziel, gespeicherte oder sich im Austausch befindliche Daten vor Einblicken von Dritten zu schützen. Wenn personenbezogene Daten durch Datendiebstahl oder technischen Pannen entwendet werden, handelt es sich um Datenpannen (Data Breaches) oder Datenlecks (Data Leaks), die den Verlust der Vertraulichkeit zur Folge haben.[20]

Die Integrität bezeichnet die Unversehrtheit von Daten, die vollständig und nicht verändert vorliegen. Eine unautorisierte Modifikation der übertragenen oder

[16] Vgl. **[MyS15]** Kap. 15, S. 2307-2310
[17] Vgl. **[Raz14]** Kap 1
[18] Vgl. **[BSI12]** S. 14
[19] Vgl. **[Müt13]** Kap. 3, S. 10-11
[20] Vgl. **[BSI16]** Kap. 1, S. 69

verarbeiteten Daten führt zum Verlust der Integrität. Die unautorisierte Veränderung von Daten kann durch Manipulation einer Webanwendung aber auch durch Vortäuschung (Spoofing) eines falschen DNS-Eintrags entstehen.[21]

Die Verfügbarkeit definiert die Nutzbarkeit eines IT-Systems bzw. von Informationen zu einem gewünschten Zeitpunkt. Der Verlust der Verfügbarkeit kann durch Überlastung eines Dienstes (z.B. durch einen Denial-of-Service-Angriff), Hard- oder Softwaredefekte an der Serverplattform oder Beschädigung der Infrastruktur im Rechenzentrum (z.B. Defekt der Stromversorgung, der Kühlung oder den Verbindungsnetzen) entstehen.[22]

Die Vertrauenswürdigkeit und der Erfolg eines digitalen Unternehmens hängen heutzutage stark von der Sicherheit der Informationen ab. Die Informationssicherheit gewinnt allgemein immer mehr an Bedeutung und wird zunehmend vom Gesetzgeber und von Kunden eingefordert. Hierzu existieren gesetzliche Vorgaben, die einen sicheren Umgang mit Daten vorschreiben.

2.2.1 Bundesdatenschutzgesetz

Das deutsche Bundesdatenschutzgesetz (BDSG) regelt den Umgang mit personenbezogenen Daten, die in Informations- und Kommunikationssystemen (digital) oder in manueller Form erhoben werden. Informationen gelten als personenbezogene Daten, wenn sie einen Personenbezug aufweisen (z.B. Name, Anschrift, Kreditkartennummer). Der „Zweck dieses Gesetzes ist es, den einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird“ (§ 1 Abs. 1 BDSG). Laut § 4 des BDSG ist die Erhebung, die Verarbeitung und die Nutzung von personenbezogenen Daten nur bei gegebener Rechtsgrundlage oder bei ausdrücklicher Zustimmung zulässig (§ 4 BDSG). Insbesondere bei Datenbanken, welche personenbezogene Daten speichern oder nutzen, sind technische und organisatorische Maßnahmen nach § 9 des BDSG zu erfüllen. Die Maßnahmen sind in der Anlage zu § 9 Satz 1 im BDSG spezifiziert und in acht technische und organisatorische Maßnahmen zum Schutz der Informationen in Datenverarbeitungsanlagen unterteilt (§ 9 BDSG). Nachfolgend werden die acht Maßnahmen des BDSG aufgezeigt.

Die Zutrittskontrolle ist die physische Beschränkung des Zutritts zu Datenverarbeitungsanlagen (z.B. Server, Datenbanksysteme), die in Unternehmen oder in Rechenzentren untergebracht sind. Maßnahmen der Zutrittskontrolle sind z.B. abgesicherte Räume, eine Personenkontrolle, ein Wachdienst oder eine

[21] Vgl. **[BSI16]** Kap. 1, S. 70
[22] Vgl. ebd. Kap. 1, S. 69

Videoüberwachung. Die Anzahl der Schutzmaßnahmen ist von der Sensibilität des Datenbestands abhängig (§ 9 Satz 1 BDSG).

Der logische Zugang zu Datenverarbeitungsanlagen wird durch Authentifizierungsverfahren (siehe Kapitel 4.2) kontrolliert. In den meisten Fällen wird die Zugangskontrolle mittels E-Mail-Adresse und Passwort oder durch die Benutzername-Passwort-Authentifizierung realisiert. Weitere Methoden sind z.B. die biometrische Benutzer-ID und der zertifikatbasierte Zugang (§ 9 Satz 1 BDSG).

Der logische Zugriff auf Ressourcen (z.B. auf personenbezogene Daten) wird mit Hilfe der Zugriffskontrolle überwacht und gesteuert. Je nach Benutzer (z.B. User, Administrator) werden vordefinierte Berechtigungen anhand der Benutzerkonten vergeben, die durch die Autorisierung geregelt werden. Es existieren zahlreiche Zugriffskontrollstrategien die in Kapitel 4.3 vorgestellt werden (§ 9 Satz 1 BDSG).

Die Weitergabekontrolle dient der logischen und physischen Datensicherheit bei der Übertragung (elektronische Weitergabe), dem Transport und der Speicherung von personenbezogenen Daten. Ziel ist die Integrität und die Vertraulichkeit während der Datenübertragung bzw. Speicherung zu gewährleisten. Daten können durch kryptographische Verfahren (siehe Kapitel 4.4) verschlüsselt gespeichert werden oder im Netzwerk verschlüsselt übertragen werden (§ 9 Satz 1 BDSG).

Die Nachvollziehbarkeit der Benutzereingaben (z.B. Einfügen, Ändern und Löschen) und die spätere Überprüfbarkeit, ist Aufgabe der Eingabekontrolle. Zur Fehlererkennung und Kontrolle werden die Eingaben jedes Benutzers in Logdateien aufgezeichnet. In der IT wird die detaillierte Protokollierung auch als Auditing bezeichnet (§ 9 Satz 1 BDSG).

Als Auftragskontrolle wird die Einhaltung von Richtlinien bei der Verarbeitung von personenbezogenen Daten in einem digitalen Auftrag bezeichnet. Dabei ist sicherzustellen, dass nur die personenbezogenen Daten, die durch die spezielle Weisung des externen Auftragsgebers beauftragt wurden, übermittelt und verarbeitet werden können (§ 9 Satz 1 BDSG).

Die Verfügbarkeitskontrolle ist mit dem Schutzziel der Informationssicherheit gleichzusetzen und muss gewährleisten, dass personenbezogene Daten vor Verlust oder logischer bzw. physischer Zerstörung geschützt sind (§ 9 Satz 1 BDSG).

Die Zwecktrennung stellt sicher, dass personenbezogene Daten, die aus unterschiedlichen Zwecken erhoben wurden, voneinander getrennt verarbeitet werden, um Überschneidungen bei der späteren Datenverarbeitung zu verhindern. Dies kann z.B. durch getrennte Ordnerstrukturen, separate Tabellen innerhalb

einer Datenbank oder durch separierte Datenbanken realisiert werden (§ 9 Satz 1 BDSG).

2.2.2 IT-Sicherheitsgesetz

Unternehmen stehen durch das BDSG in der Pflicht, angemessene Sicherheitsmaßnahmen zum Schutz personenbezogener Daten zu implementieren. Das neue IT-Sicherheitsgesetz verschärft die Anforderungen für die Informationssicherheit deutscher Unternehmen und den Betreiber kritischer Infrastrukturen. Das Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme (IT-Sicherheitsgesetz) ist am 25. Juli 2015 durch die deutsche Bundesregierung beschlossen worden. Das Gesetz richtet sich in erster Linie an Betreiber von kritischen Infrastrukturen. Darunter fallen Infrastrukturen aus den Bereichen Energie, Wasser, Verkehr, Informationstechnik, Telekommunikation, Transport, Gesundheit, Ernährung sowie Finanz- und Versicherungswesen, da deren Ausfall oder Störung zu enormen Engpässen und Gefahren für die öffentliche Sicherheit führt (§ 1 Abs. 2 IT-Sicherheitsgesetz). Das Gesetz fordert angemessene Vorkehrungen, um den Schutz von Telekommunikations- und Datenverarbeitungssystemen zu gewährleisten. Betreiber kritischer Infrastrukturen haben Störungen der Vertraulichkeit, Verfügbarkeit, Integrität und Authentizität unverzüglich dem Bundesamt für Sicherheit in der Informationstechnik (BSI) zu melden (§ 11 Abs. 1c IT-Sicherheitsgesetz). Die Authentizität steht für ein untergeordnetes Schutzziel der Informationssicherheit und ergibt sich aus der Echtheit, der Überprüfbarkeit und der Vertrauenswürdigkeit von Informationen.

Größeres Ausmaß als früher hat das neue IT-Sicherheitsgesetz für Betreiber von Webanwendungen wie z.B. Online-Shops. Die im Rahmen des technisch Möglichen und wirtschaftlich Zumutbaren sicherstellen müssen, dass durch technische und organisatorische Maßnahmen kein unerlaubter Datenzugriff möglich ist. Konkret bezieht sich das Gesetz auf die Verletzung personenbezogener Daten sowie gegen Störungen durch äußere Angriffe (§ 13 IT-Sicherheitsgesetz). Das Gesetz verschärft die Regeln zum Schutz personenbezogener Daten und nimmt dabei alle Betreiber von Internetauftritten und angebotenen Internetdiensten (z.B. Youtube) in die Pflicht. Es gibt keine Unterscheidung zwischen professionellen Unternehmen oder privaten Internetauftritt.

2.2.3 Standards und Normen

Sicherheitsstandards bieten etablierte Lösungswege zur Absicherung von informationstechnischen Systemen (IT-System) und konkretisieren die verbindlichen Anforderungen der Gesetzgebung. Die Standards sind in Normen unterteilt, die durch spezielle Gremien weiterentwickelt werden. Best Practices

stellen ebenfalls eine Art Standard dar, sofern es keine anwendungsspezifischen Normen zu berücksichtigen gilt.[23] Die Einhaltung der Anforderungen und Richtlinien wird durch Audits (Überprüfungsverfahren) analysiert und zertifiziert. Welche Standards oder Normen ein Unternehmen einzuhalten hat, ist von der Branche, dem Geschäftsmodell sowie den Ansprüchen Dritter abhängig. Die bekanntesten Standards und Normen nach dem „Kompass der IT-Sicherheitsstandards“ von BITKOM aus dem Jahr 2013 sind:[24]

- ISO/IEC 27000 (Informationssicherheits-Managementsysteme)
- IT-GS (BSI IT-Grundschutz)
- PCI DSS (Payment Card Industry Data Security Standard)
- Cobit (Control Objectives for Information and Related Technology)
- ITIL (Information Technology Infrastructure Library)
- ITSEC/CC (Information Technology Security Evaluation Criteria/Common Criteria)
- IDW PS 330 (Abschlußprüfung bei Einsatz von Informationstechnologie)
- OCTAVE (Operationally Critical Threat, Asset and Vulnerability Evaluation)
- SDM (Standard-Datenschutzmodell)[25]

Mit der Zertifizierung und Anerkennung eines Standards bzw. einer Norm kann die Qualität gesichert, Transparenz geschaffen und Verbraucher geschützt werden. Weiterhin wird durch die Erfüllung und Einhaltung der IT-Sicherheitsstandards die Angriffsfläche von potenziellen Bedrohungen und Sicherheitsrisiken minimiert.[26]

[23] Vgl. **[Sch10]** Kap. 2.6.1, S. 32-33
[24] Vgl. **[BIT15]** S. 9-12
[25] Vgl. ebd. S. 9-12
[26] Vgl. **[Bun16]**

3 Bedrohungen und Sicherheitsrisiken

In Kapitel 3 werden Sicherheitsrisiken vorgestellt, die an die Studie „10 Most Common Security Vulnerabilities in Enterprise Databases“ aus dem Jahre 2013 angelehnt sind.[27] Die aufgeführten Bedrohungen greifen die zentralen Aspekte der Datenbanksicherheit auf und dienen der Ableitung von Sicherheitsmaßnahmen.

3.1 Sicherheitsrelevante Fehlkonfiguration

Viele Datenbanksysteme sind nach einer Neuinstallation nicht ausreichend sicher konfiguriert. Dies ist in erster Linie der mitgelieferten Standardkonfiguration (Default Configuration) geschuldet. Die Installationsroutine verfügt über Standardprozeduren, die Schwachstellen enthalten können. Dazu gehören Standardpasswörter, eine vorinstallierte Testdatenbank, ein schwaches Berechtigungs- bzw. Zugriffschutzkonzept oder fehlende Schutzfunktionen (z.B. Datenverschlüsselung).[28] Weiterhin können Funktionserweiterungen standardmäßig aktiviert sein, die für den vorgesehenen Verwendungszweck überflüssig sind, oder Zusatzfunktionen, die nach einer erfolgreichen Installation erst noch manuell konfiguriert (z.B. Übertragungssicherheit) werden müssen. Zur Risikominimierung kann die Default-Konfiguration manuell oder automatisch angepasst werden. Die automatische Anpassung mittels Skripten oder Anwendungen stellt die sicherste Alternative dar. Eine abgesicherte Konfiguration sowie sorgfältig ausgewählte Sicherheitseinstellungen und Erweiterungen sind die Grundlage für einen sicheren Betrieb des Datenbanksystems.[29]

3.2 Unsichere Übertragung

Bei der Kommunikation über das Internet kann die Übertragung durch unsichere Protokolle manipuliert bzw. mitgelesen werden. Bei einer unsicheren Übertragung werden die Nutzdaten im Klartext übermittelt. Hierzu zählt z.B. das Hypertext Transfer Protocol (HTTP).[30] Durch Man-in-the-Middle Angriffe kann die Kommunikation zwischen Client und Server mitgeschnitten und so z.B. die Benutzerdaten während der Authentifizierung ausspioniert werden. Die Übertragungssicherheit ist ein wesentlicher Aspekt der Datenbanksicherheit. Das Verschlüsselungsprotokoll ist optional und in der Standardkonfiguration des Datenbankmanagementsystems nicht installiert. Übertragungssicherheit kann durch die Transportsicherheitsschicht (Transport Layer Security, kurz TLS) oder durch das veraltete Verschlüsselungsprotokoll (Secure Sockets Layer, kurz SSL) erzielt werden. Zur Nutzung der Übertragungssicherheit müssen Zertifikate und Schlüssel server- und clientseitig generiert werden und in die Konfigurationsdatei

[27] Vgl. **[Lan13]**
[28] Vgl. **[Lan13]** S. 5-6
[29] Vgl. **[OWASP]** S. 12
[30] Vgl. **[BSI16]** Kap. G 2.87, S. 622-623

des Datenbanksystems eingebettet werden. Je nach verwendetem Datenbanksystem muss ggf. die Verschlüsselungssoftware (z.B. OpenSSL) noch nachträglich installiert werden. Weitreichende Risiken bei der Übertragungssicherheit stellen abgelaufene Zertifikate, ein zu schwach gewählter Schüssel oder die Verwendung des veralteten und unsicheren SSL-Standards dar.[31]

3.3 Schwache Authentifizierung

Bei der Neuinstallation eines Datenbanksystems werden automatisch Standardbenutzernamen vergeben. Bei MySQL wurde in früheren Versionen der Benutzer „root“ ohne Passwort, aber mit sämtlichen Privilegien am Datenbanksystem, erzeugt. Der Zugang zum Datenbanksystem war zwar nur Server intern (Localhost bzw. lokale IP-Adresse 127.0.0.1) möglich, jedoch konnten durch weitere Sicherheitslücken (Vulnerabilities) diese Schwachstelle ausgenutzt werden. Bei anderen Datenbankherstellern wird ein Standardbenutzername (z.B. Oracle SYS oder SYSDBA) mit selbst gewählten Passwort generiert. Wird der Standardbenutzername des privilegierten Benutzerkontos (Account) nicht geändert, stellt das Datenbanksystem ein leichtes Ziel für Brute-Force- oder Dictionary Attack dar. Bei der Brute-Force-Methode wird das Passwort durch Austesten aller Zeichenkombinationen bis zu einer festgelegten Zeichenlänge ausspioniert. Beim Wörterbuchangriff (Dictionary Attack) werden vordefinierte Listen mit oft genutzten Passwörtern ausprobiert.[32] Die Sicherheit der Authentifizierung hängt von vielen einzelnen Faktoren ab. Hierzu zählt das verwendete Authentifizierungsverfahren, die Passwort- und Zertifikatstärke, der benutzte Übertagungs-/Verschlüsselungsalgorithmus, der Mehrfachschutz vor Falscheingaben sowie die zeitlich limitierte Passwortgültigkeit.[33]

3.4 Mäßige Zugriffskontrolle

Die Zugriffskontrolle überprüft die Berechtigungen, die Benutzer an einem Datenbanksystem nach einer erfolgreichen Authentifizierung erhalten. Sie regelt wer wie und auf welche Ressourcen zugreifen darf. Dazu werden die Datenobjekte (z.B. Tabellen, Datenbankprozeduren) durch Zugriffsrechte (Grants) geschützt. Ziel der Zugriffskontrolle ist der Schutz der Vertraulichkeit und der Integrität von sensiblen Daten. Fehlerhafte Einstellungen in der Zugriffskontrolle aber auch Programmierfehler im dem Datenbanksystem haben eine nicht autorisierte Rechteausweitung (Privilege Escalation) zur Folge. Nachstehende Maßnahmen erhöhen die Sicherheit im Bereich der Zugriffskontrolle: privilegierte Konten zu

[31] Vgl. **[Lan13]** S. 7-8
[32] Vgl. **[BSI16]** Kap. G5.18, S. 1117
[33] Vgl. **[OWASP]** S. 9

separieren und Benutzerkonten für spezielle Funktionen (z.B. Verwaltung, Backups, Löschen und Hinzufügen von Funktionen) anzulegen. Dadurch kann der potenzielle Schaden am Datenbanksystem durch Missbrauch der Zugangsdaten privilegierter Benutzerkonten minimiert werden.[34]

3.5 Fehlende Verschlüsselung

Anmeldeinformationen wie auch personenbezogene Kundendaten (z.B. Postanschrift) sollten ausschließlich unter Verwendung von Verschlüsselungsverfahren in einer Datenbank gespeichert werden. Bei einer Datenpanne ist eine ausreichende Verschlüsselung der Daten (z.B. Kreditkarteninformationen) die letzte Sicherheitsinstanz. Fehlende oder unzureichend verschlüsselte Daten sind die häufigste Ursache für unautorisierte Veröffentlichungen. Datenbanksysteme verfügen über Verschlüsselungsverfahren (z.B. Advanced Encryption Standard) um Kundendaten sowie Passwörter zu schützen. Die Sicherheit der gespeicherten Daten hängt stark von dem verwendeten Verschlüsselungsalgorithmus und der Länge des eingesetzten Schlüssels (Bitlänge) ab. Die Verschlüsselung bildet das grundlegende Fundament der Datenbanksicherheit. Verschlüsselungsverfahren können durch Implementierungsfehler, unsichere Verfahren, Nachlässigkeit bei der Aufbewahrung des Schlüssels oder durch Datenbankerweiterungen, die das Passwort defaultmäßig in Klartext speichern, ausgehebelt werden.[35]

3.6 Sicherheitslücken

Datenbanksysteme wie auch alle anderen Anwendungen im Internet sind durch mangelndes Patch-Management verwundbar. Die Sicherheitslücken werden meistens durch Programmierfehler in der Software hervorgerufen. Es entstehen Schwachstellen und Sicherheitsmechanismen können umgangen werden.[36] Dies ermöglicht einen unautorisierten Zugriff auf das betroffene System. Im Folgenden werden einige Schwachstellen genannt:

- Speicherüberlauf (Buffer Overflow)
- Rechteausweitung
- Code-Ausführung (Code Execution)
- Einschränkung der Verfügbarkeit von Diensten
- Umgehung der Benutzerkennung (Authentication Bypass)
- Missbrauch von Zugriffsrechten auf Dateien (File Privilege Abuse)

[34] Vgl. **[Lan13]** Kap. 6, S.10
[35] Vgl. **[OWASP]** S. 13
[36] Vgl. ebd. S. 16

Des Weiteren existieren Zero-Day- (unentdeckte Sicherheitslücken, die der Öffentlichkeit nicht bekannt sind) und Less-Than-Zero-Day-Sicherheitslücken (unveröffentlichte Sicherheitslücken, die nur einem bestimmten Personenkreis bekannt sind). Diese können Dienste und Datenbanksysteme verwunden, ohne dass es einen effektiven Schutz oder ein Anzeichen für einen Softwarefehler bzw. eine Schwachstelle gibt.[37] Dramatischer ist die Tatsache, dass Monate vergehen bis Softwarehersteller Sicherheitsupdates zum Schließen der Lücken entwickeln und der Öffentlichkeit bereitstellen.[38]

[37] Vgl. **[Poh]** S. 1
[38] Vgl. **[Kea14]** Kap. 2, S. 11

4 Aspekte der Datenbanksicherheit

In Kapitel 4 werden die wesentlichen Aspekte der Datenbanksicherheit dargestellt. Diese leiten sich aus den in Kapitel 3 dargestellten Bedrohungen und Sicherheitsrisiken von Datenbanksystemen ab. Die Sicherheitsmechanismen der Datenbanksicherheit sind in Konfiguration, Authentifizierung, Zugriffskontrolle, Verschlüsselung, Auditing, Sicherheitsupdate und Sicherheitsvorfälle unterteilt. Die Wichtigkeit dieser sieben Aspekte wird durch das IBM Whitepaper „Acht Schritte zur ganzheitlichen Datenbanksicherheit“ gestützt.[39]

4.1 Konfiguration

Die Konfiguration der Datenbanksoftware findet immer erst nach einer erfolgreichen Installation statt. Die Installation eines Datenbanksystems kann durch eine grafische Installationsroutine aber auch durch Konsolenaufrufe initialisiert werden. Während der Installation werden unterschiedliche Schutzmaßnahmen konfiguriert und die Standardkonfiguration erzeugt. In den meisten Fällen werden hierbei Passwörter für den administrativen Zugang eingerichtet und eine Testdatenbank angelegt. Je nach Hersteller kann die Datenbankkonfigurationsdatei durch eine geführte grafische Konfigurationsroutine oder auch noch manuell editiert werden.[40] Vor dem Freischalten bzw. Online gehen (Internet) des Datenbanksystems müssen sicherheitsrelevante Aspekte entsprechend dem vorgesehenen Anwendungszweck sicher konfiguriert sein. Die wichtigsten Schutzmechanismen der Konfigurationssicherheit bei Datenbanksystemen sind:

- Standardbenutzernamen und -passwörter ändern
- Unnötige Erweiterungen deaktivieren
- Verbindungsverschlüsselung aktivieren
- Testdatenbank löschen
- Überwachung und Protokollierung aktivieren
- Externe Dienste deaktivieren

4.2 Authentifizierung

Bei einem Anmeldeversuch an einem Datenbankmanagementsystem wird der Zugriff auf den Datenbankdienst durch die Authentifizierung überprüft. Die Authentifizierung verlangt dafür die Vorlage eines Nachweises, um den Benutzer oder die Anwendung an dem Datenbankmanagementsystem zu autorisieren. Dafür wird die Echtheit der vorgegebenen Identität des Kommunikationspartners

[39] Vgl. **[Nat10]**
[40] Vgl. **[Gre91]** Kap. 3.1, S. 65

überprüft.[41] Der Zugang zu Datenbankmanagementsystemen lässt sich durch unterschiedliche Authentifizierungsmethoden vor unautorisierten Zugriffen schützen.

4.2.1 Password Authentication Protocol

Password Authentication Protocol (PAP) – auch Benutzername-Passwort-Authentifizierung ist das gängigste Authentifizierungsverfahren. Der Server, in diesem Falle das Datenbankmanagementsystem, bietet einen Dienst (Service) an, mit dem sich die Clients (z.B. Benutzer) authentifizieren können.[42] Die Benutzerdaten (Benutzername, Passwort, oft auch noch Hostname oder IP-Adresse) befinden sich in der Benutzertabelle der Systemdatenbank. Das Passwort wird durch einen Hash-Algorithmus verschlüsselt. Die Kommunikationsbeziehung zur Authentifizierung wird durch den Client initialisiert. Dazu wird in der Eingabeaufforderung (Konsole) oder Weboberfläche der Host, der Benutzername, das Passwort und die Datenbank angegeben. Im nächsten Authentifizierungsschritt überprüft der Server die Benutzereingaben und vergleicht den aus dem eingegebenen Passwort generierten Hashwert mit dem in der Benutzertabelle.

4.2.2 Pluggable Authentication Modules

Die Pluggable Authentication Modules (PAM) ist ein Linux-basiertes Authentifizierungsverfahren, das über das gesamte Netzwerk hinweg eine einheitliche Benutzererkennung ermöglicht. Dies wird durch einen zentralen Authentifizierungsdienst realisiert, welcher eine API zur Verfügung stellt. Hierzu muss das Datenbanksystem über die PAM-Schnittstelle verfügen. Der Vorteil ist, dass alle Dienste (z.B. Datenbankmanagementsystem, Secure Shell, File Transfer Protocol) einer Server-Plattform sich zentral durch eine Authentifizierung verwalten lassen. Jedoch ist ein zentrales Passwort für alle Serverdienste aus sicherheitstechnischen Gründen unerwünscht.[43]

4.2.3 Public-Key-Infrastructure

Bei der Public-Key-Infrastructure (PKI) erfolgt die Authentifizierung durch digitale Zertifikate (z.B. X.509v3). Ein Benutzer meldet sich unter Verwendung eines Schlüsselpaares, bestehend aus einem geheimen (privaten Schüssel) und einem öffentlichen Schüssel, an.[44] Dabei ist der öffentliche Schlüssel auf dem Server hinterlegt und der private Schüssel befindet sich im Besitz des Benutzers. Die Kommunikationspartner brauchen hier im Vergleich zur Benutzername-Passwort-Authentifizierung den geheimen Schüsseln nicht zu kennen. Das Schüsselpaar aus

[41] Vgl. **[BSI16]** Kap. 4, S. 98
[42] Vgl. **[RFC1334]** S. 3-4
[43] Vgl. **[Mor10]** Kap. 1-3, S. 1-4
[44] Vgl. **[BSI]**

geheimen und privaten Schlüssel wird durch das asymmetrische kryptographische Verfahren Rivest, Shamir und Adleman (RSA) generiert.[45]

4.2.4 Lightweight Directory Access Protocol

Das Lightweight Directory Access Protocol (LDAP) ist ein TCP/IP basierter Verzeichnisdienst (Directory Service). Die Kommunikation beruht auf dem Client-Server-Modell und ermöglicht den Austausch von Daten zwischen LDAP-Client und Directory-Server (X.500 Standard).[46] Der Verzeichnisdienst besitzt eine baumartige Struktur und ermöglicht das Abfragen von Einträgen. Die Authentifizierung mittels LDAP wird durch das Simple Authentication and Security Layer (SASL) Framework realisiert.[47] Das Framework verfügt über eine Abstraktionsschicht, wodurch Protokolle (z.B. LDAP) und Authentifizierungsmechanismen (z.B. PLAIN) verknüpft werden. Die Sicherheitsebene (Security Layer) wird durch die Transportschichtsicherheit (TLS) erreicht.[48]

4.2.5 Kerberos

Kerberos ist ein dezentraler Authentifizierungsdienst, der sich nicht direkt am Server (z.B. Datenbankmanagementsystem) authentifiziert, sondern an einem sicheren dritten Kerberos-Server (Key Distribution Center). Kerberos unterstützt in der Version 5 den Single Sign-on Mechanismus, der eine einmalige Authentifizierung aller Serverdienste per Tickets erlaubt.[49] Zur Authentifizierung fordert der Client am Kerberos-Server ein Ticket Granting Ticket (TGT) an. Das Ticket bestätigt die Authentifizierung am Kerberos-Server und dient zur Anforderung weiterer Service Tickets ohne erneute Anmeldung. Anschließend wird ein Service Ticket angefordert, durch den Ticket Granting Service (TGS) des Kerberos-Servers ausgestellt und dem Client übermittelt. Mit dem ausgestellten Service Ticket erfolgt die Authentifizierung am Serverdienst (z.B. Datenbankmanagementsystem).[50] Vorteil der Methode ist, dass der Client, der Server und der Kerberos-Server sich gegenseitig über symmetrische Schlüssel verifizieren und die Echtheit des Kommunikationspartners identifizieren.

[45] Vgl. **[RFC2459]** Kap. 1-3; S. 5-9
[46] Vgl. **[RFC4510]** S. 1-7
[47] Vgl. **[RFC4513]** S. 1
[48] Vgl. **[RFC4422]** S. 2
[49] Vgl. **[RFC5021]** S. 1-3
[50] Vgl. **[RFC1510]** Kap. 1, S. 5-7

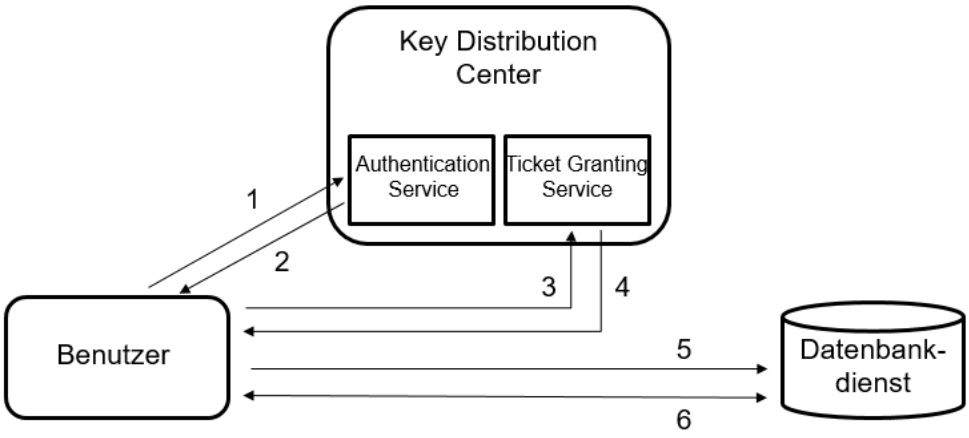

Abbildung 2 - Kerberos Authentifizierung[51]

Die Funktionsweise der Kerberos Authentifizierungsmethode lässt sich in einem „Client-Server-Modell“ beispielhaft darstellen (siehe Abbildung 2):

1. Anforderung des Ticket Granting Tickets
2. Erhalt des Ticket Granting Ticket
3. Service Ticket anfragen
4. Service Ticket erhalten
5. Anforderung des Datenbankdienstes durch Service Ticket
6. Datenbankdienst benutzen

4.2.6 Remote Authentication Dial-In User Service

Der Remote Authentication Dial-In User Service (RADIUS) ist ein Client-Server-Sicherheitsprotokoll, das neben der Authentifizierung auch die Autorisierung (Authorization) und Abrechnung (Accounting) unterstützt.[52] Auch bekannt als triple A (AAA). Die Autorisierung überprüft nach einer erfolgreichen Authentifizierung, ob Benutzer berechtigt (autorisiert) sind einen Befehl (z.B. löschen einer Datenbank) abzusetzen. Autorisierung stellt einen alternativen Begriff für die Zugangskontrolle dar. Die Abrechnung diente früher Internetdienstanbieter (Internet Service Provider) für die Erfassung und Abrechnung von erbrachten Leistungen. Das Radius-Protokoll wurde oft von Internetdienstanbietern genutzt, um Einwahlverbindungen (z.B. Modem, DSL) zu authentifizieren. Heutzutage wird das Sicherheitsprotokoll für die Zugangskontrolle von Wireless LANs in Firmennetzwerken und für die Authentifizierung von Anwendungen verwendet. Der Client, sprich der Benutzer, meldet sich mit seiner Benutzername-Passwort-Kennung am Datenbankmanagementsystem an. Das Datenbankmanagementsystem agiert dabei als Radius-Client, der eine Zugangsanfrage an den Radius-Server weiterleitet. Client und Server vereinbaren

[51] Angelehnt an **[Neu94]**
[52] Vgl. **[RFC2865]** S. 1-3

vorab einen sicheren Schlüssel (kryptische Zeichenfolge), um Informationen zu verschlüsseln und an dem Radius-Server zu authentisieren. Daraufhin akzeptiert der Radius-Server die Authentifizierungsanfrage oder lehnt sie ab. Zur Überprüfung der Korrektheit der Daten und des Absenders (Radius-Server) berechnet der Server eine kryptographische Hashfunktion aus der Zugangsanfrage, die dem Client mit Antwort auf die Authentifizierungsanfrage übermittelt wird.[53]

4.2.7 Secure Socket Layer

Der Secure Sockets Layer ist ein standardisiertes Verschlüsselungsprotokoll, das ursprünglich von Netscape Communications Corporation entwickelt wurde, um Netzwerk- und Internetverbindungen abzusichern. Die Hauptaufgabe des SSL-Protokolls ist die Übertragungssicherheit zwischen zwei Kommunikationspartner sicherzustellen. Dazu ist das SSL-Protokoll in zwei Schichten aufgebaut. Dem SSL Record Protocol, das direkt auf das Transportprotokoll aufsetzt, und dem darüber liegenden SSL Handshake Protocol. Das SSL Handshake Protocol wird zur Datenverschlüsselung (Data Encryption) und zur Authentifizierung verwendet.[54] Im SSL-Protokoll legt die Cipher Suite fest, welche Algorithmen zum Aufbau der gesicherten Datenverbindung verwendet werden. Die Cipher Suite verfügt über eine Kombination aus vier Funktionen: Schlüsselaustausch (z.B. RSA), Authentifizierung (z.B. RSA), Hashfunktion (z.B. Message-Digest Algorithm 5) und die Verschlüsselung (z.B. Data Encryption Standard).[55]

4.2.8 Two-Factor Authentication

Two-Factor Authentication (2FA) ist eine Authentifizierungsmethode, bei der die Überprüfung der Identität in zwei Schritten (z.B. Bankkonto) durch eine zweite Sicherheitsebene erfolgt. Neben der anfälligen Benutzername-Passwort-Authentifizierung wird eine weitere, unabhängige Identitätsprüfung durchgeführt. Diese Überprüfung erfolgt über einen zweiten unabhängigen Weg, z.B. über den Kurznachrichtendienst (SMS), eine E-Mail oder die Anwendungssoftware (App).[56]

4.3 Zugriffskontrolle

Die Zugriffskontrolle (Access Control) beschreibt ein IT-Sicherheitskonzept, das reguliert, wer und wie auf eine Tabelle bzw. Sichten innerhalb einer Datenbank zugreifen darf. Das Ziel der Zugriffskontrolle ist es, die Datenbank vor unautorisierten und unberechtigten Zugriffen zu schützen. Der Zugriffsschutz wird

[53] Vgl. **[Els]**
[54] Vgl. **[RFC6101]** Kap. 1, S. 4
[55] Vgl. ebd. Kap. 6, S 47- 50
[56] Vgl. **[Sec14]**

durch die folgenden Zugriffskontrollstrategien (Access Control Policies) sichergestellt.[57]

4.3.1 Discretionary Access Control

Die benutzerbestimmbare Zugriffskontrolle (Discretionary Access Control kurz DAC) regelt auf Basis von Benutzeridentitäten und Berechtigungsregeln (Authorization Rules) den Zugriff auf Datenbankebene. Die Berechtigungen bestimmen dabei die Zugriffsregeln, z.B. welcher Benutzer wie auf Ressourcen innerhalb der Datenbank zugreifen darf. Die Zugriffsrechte (Access Rights) der benutzerbestimmbaren Zugriffskontrolle können als Relationen von Objekten, Subjekten und Rechten beschrieben werden. Die Rechtezuweisung muss einzeln für jeden Benutzer erfolgen. In der grundlegenden Form wird die Zugriffskontrolle durch eine Zugriffskontrollmatrix (Access Control Matrix) verwaltet. Dabei darf das Subjekt (z.B. ein Benutzer) das Objekt (z.B. eine Tabelle) genau dann lesen, wenn read ϵ M(s,o) entspricht (siehe dazu Tabelle 1).[58]

Subjekte/ Objekte	Datei 1	Datei 2	Datei 3	…
Armin	r,w,x	-	-	
Ute	r	-	r,w	
Ralf	-	r	-	

M(Armin,Datei1)=(r,w,x)

Tabelle 1 - Zugriffskontrollmatrix[59]

Grundsätzlich wird die Zugriffskontrolle nur auf den zu schützenden Objekten durchgeführt und steht dadurch vollkommen unter der Kontrolle des Eigentümers. Da neben den Objekten und Subjekten keine weitere Instanz den Zugriff von Subjekten auf Objekten überwacht, ist das Zugriffsmodell anfällig für Rechteausweitungen.[60] Die Struktur der benutzerbestimmbaren Zugriffskontrolle ist auch nur für kleine Nutzerzahlen geeignet.

4.3.2 Mandatory Access Control

Bei der regelbasierten Zugangskontrolle (Mandatory Access Control kurz MAC) wird die Regulierung der Zugriffsberechtigung nicht nur auf Grundlage der Benutzeridentität und des Objekts wie bei der DAC umgesetzt, sondern um allgemeine Regeln und Eigenschaften erweitert. Die Regeln und Eigenschaften bilden zusammen den Sicherheitskontext. Jedes Objekt und Subjekt ist dabei einer Sicherheitsklasse zugeordnet, die sich aus Schutzstufen und einer Reihe von

[57] Vgl. **[Opp13]** Kap. 5.2, S. 68-74
[58] Vgl. **[Jav08]** Kap. 2.1.1, S. 14
[59] Angelehnt an **[But71]**
[60] Vgl. **[Sch16]**

Regeln zusammensetzt. Die regelbasierten Zugriffskontrollen sind in Multi-Level-Sicherheitssysteme (Multi-Level Security) nach der Vertraulichkeit und in multilaterale Sicherheitsmodelle (Multilateral Security) nach der Integrität unterteilt.[61]

Das Multi-Level-Sicherheitssystem ist die bekannteste Zugangskontrolle, die auf dem Bell-LaPadula-Modell beruht. In diesem Sicherheitssystem sind die Schutzstufen immer vertikal angeordnet und nach Objekten unterteilt. Die Subjekte (Benutzer) sind immer einer Schutzstufe zugeordnet. Objekte und Subjekte werden dabei in Schutzstufen nach der Vertraulichkeit untergliedert. Die Schutzstufen des Objekts werden auch als Klassifizierung (Classification) und die Schutzstufe des Subjekts als Freigabe (Clearance) bezeichnet. Die Zugangskontrolle wird durch die Top-Down- und Bottom-Up-Informationsflusskontrolle realisiert.[62] Dabei existieren folgende Regeln:

> **No-Read-Up** – Ein Subjekt kann nur ein Objekt lesen, wenn die Schutzstufe des Subjekts höher ist als die des Objekts.[63]
>
> **No-Write-Down** – Ein Subjekt kann auf ein Objekt nur dann schreiben, wenn die Schutzstufe des Objekts höher ist als die des Subjekts.[64]

Das multilaterale Sicherheitsmodell erweitert das Multi-Level-Sicherheitssystem um ein horizontales Zugriffssystem (Integrity Class). Das Neben der vertikalen Klassifizierung der Objekte und Freigaben der Subjekte wird eine weitere horizontale Sicherheitskontrolle durch Kennwörter (Labels) implementiert.[65] Die zusätzliche Kontrolle durch die Integritätsklasse verhindert, dass Subjekte indirekt Dateien verändern ohne über berechtigten Schreibzugriff zu verfügen. Hierfür gelten die Regeln:

> **No-Read-Down** – Ein Subjekt kann nur ein Objekt lesen, wenn die Integritätsklasse des Objekts höher ist als die Integritätsklasse des Subjekts.[66]
>
> **No-Write-Up** – Ein Subjekt kann auf ein Objekt nur dann schreiben, wenn die Integritätsklasse des Subjekts höher ist als die Integritätsklasse des Objekts.[67]

[61] Vgl. **[Sol08]** Kap. 7.3.3, S. 279
[62] Vgl. **[Spe08]** Kap. 2.1.3, S. 34-35
[63] Vgl. **[Jav08]** Kap. 2.1.1, S. 15
[64] Vgl. ebd. Kap. 2.1.1, S. 15
[65] Vgl. **[Ros08]** Kap. 9.2, S.279
[66] Vgl. **[Jav08]** Kap. 2.1.1, S. 15
[67] Vgl. ebd. Kap. 2.1.1, S. 15

Die regelbasierten Zugriffskontrollen bestehend aus Multi-Level-Sicherheitssysteme und multilaterale Sicherheitsmodelle ergänzen sich gegenseitig, um die Vertraulichkeit und die Integrität zu schützen.[68]

4.3.3 Role-Based Access Control

Bei der rollenbasierten Zugriffskontrolle (Role-Based Access Control kurz RBAC) werden die Zugriffsrechte nicht unmittelbar an Subjekte gebunden, sondern mit Rollen (Roles) verknüpft, die im Datenbanksystem hierarchisch unterteilt sind. Der Datenbankadministrator steht an der Spitze der Hierarchie und ermöglicht das Anlegen (Create) von Rollen oder Benutzern. Absteigend sind die Zugriffsberechtigungen wie bei einer Pyramide strukturiert. Jeder Benutzer kann einer oder mehreren Rollen gleichzeitig zugeordnet werden. Die Rollen vererben ihre Zugriffsrechte an alle zugehörigen Benutzer der Rolle. Die Zugriffsrechte der jeweiligen Rollen können durch den Datenbankadministrator vergeben werden oder sind schon in der Standardkonfiguration des Datenbanksystems enthalten. Durch die dreistufige Gliederung in Benutzer, Rollen und Benutzergruppen ist es möglich die Benutzer über Rollen- und Gruppenzuordnung zu kontrollieren. Eine Gruppe erleichtert zusätzlich die Zuweisung von Rollen an Benutzern. Die Aufgaben der rollenbasierten Zugriffskontrolle sind die Zuordnung der Benutzer-Rollen, den Rollenbeziehungen und den Rollenberechtigungen. Dadurch ist es möglich, einen großen Benutzerstamm auf einfache Weise zu administrieren.[69]

4.3.4 Label-Based Access Control

Die Label-Based Access Control (LBAC) ist eine Zugriffssteuerung auf Zeilenebene. Die Sicherheitskennsätze (Security Labels) werden auf jede Datenzeile vergeben und steuern so die Zugriffe auf Zeilenebene. Die Sicherheitskennsätze lassen sich individuell anhand der Anforderungen und der Sicherheitsrichtlinien konfigurieren. Dadurch lassen sich Zugriffsrichtlinien direkt auf Zeilenebene anwenden und Berechtigungen zum Lesen sowie Schreiben jedem Benutzer einzeln zuweisen. Dies ermöglicht die Zugriffskontrolle bis hin zur einzelnen Informationszeile. Die kennsatzbasierte Zugriffskontrolle lässt sich zusätzlich zur benutzerbestimmbaren Zugriffskontrolle einsetzen und ermöglicht dadurch eine fein granulierte Zugriffssteuerung.[70]

4.3.5 Code-Based Access Control

Die Code-Based Access Control (CBAC) ist Teil der Sicherheit von Laufzeitumgebungen. Laufzeitumgebung (Common Language Runtime) bezeichnet eine virtuelle Ausführungsumgebung, die das Schreiben und das

[68] Vgl. **[Ger08]** Kap. 1, S. 6-8
[69] Vgl. **[Spe08]** Kap. 2.1.5, S. 37
[70] Vgl. **[IBM]**

Ausführen von Quellcode ermöglichen wie z.B. der Procedural Language/Structured Query Language (PL/SQL) die eine proprietäre Programmiersprache bei Datenbanksystemen darstellt. Diese Laufzeitumgebungen erleichtern den Entwicklungsprozess neuer Funktionen oder Routinen für das Datenbanksystem. Falls ein vertrauenswürdiger Programmcode (Trusted Code) einen nicht vertrauenswürdigen Programmcode (Untrusted Code) aufruft, können sich die Berechtigungen verändern. Dabei kann der nicht vertrauenswürdige Programmteil (z.B. Datenbankbenutzer) die Zugriffsrechte (Permissons) des ausführenden Benutzers erben. Die Aufgabe des CBAC ist es, den Zugriff nach den Sicherheitsbestimmungen und Rechtekontext der jeweiligen Anwendung zu wahren.[71]

4.4 Verschlüsselung

Sensible Daten können in einem Datenbanksystem verschlüsselt abgespeichert werden und sind somit ohne Kenntnis des eingesetzten Verschlüsselungsverfahrens und der genutzten Schlüssel nicht mehr lesbar. Das Verschlüsselungsverfahren wandelt unter Verwendung eines Schlüssels lesbare Informationen (Klartext) in einen Geheimtext um. Umgekehrt kann der Geheimtext wieder nur mit Hilfe des Schlüssels in einen lesbaren Zustand überführt werden. Zur Verschlüsselung können das asymmetrische und das symmetrische Verfahren eingesetzt werden. Das asymmetrische Verfahren wurde im Kapitel 4.2.3 Public-Key-Infrastructure beschrieben. Das symmetrische Verfahren verwendet für die Ver- und Entschlüsselung denselben (identischen) Schüssel. Die symmetrischen Verschlüsselungsverfahren unterteilen sich in die Blockverschlüsselung (Block Cipher) und Stromverschlüsselung (Stream Cipher).[72]

4.4.1 Blockverschlüsselung

Mit der Blockverschlüsselung werden Klartextblöcke in Geheimtextblöcke abgebildet. Dies erfolgt in fest vorgeschriebenen Blocklängen und in mehreren Runden. Die ersten Verschlüsselungsalgorithmen der Blockverschlüsselung sind nach dem Feistel-Netzwerk aufgebaut (siehe Abbildung 3).

[71] Vgl. **[Gol11]** Kap. 20.4, S. 393
[72] Vgl. **[Ern15]** Kap. 4, S. 137

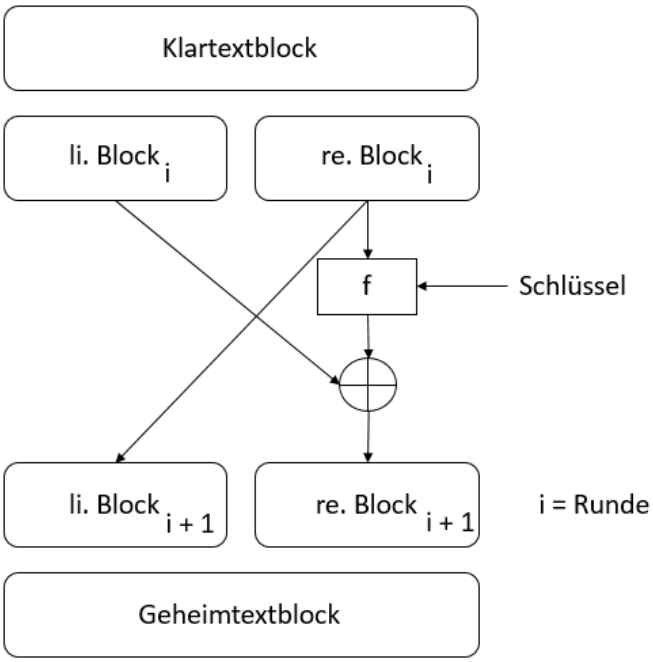

Abbildung 3 - Feistel-Netzwerk[73]

Der zusammenhängende Klartext wird zuerst in Klartextblöcke mit üblicherweise 64-Bit Länge (8 Bytes) zerlegt. Falls die Länge des Klartextes nicht die Länge des Blocks übersteigt, werden nach einem festgelegten Verfahren die restlichen Bits aufgefüllt (Padding). Jeder Klartextblock aus 64-Bit wird anschließend in zwei (linke und rechte) Blockhälften mit je 32-Bit Länge für die erste Verschlüsselungsrunde aufgeteilt. Danach werden die zwei Blockhälften über mehrmaliges Durchlaufen (Runden) mit nachfolgend beschriebenem Verfahren verschlüsselt. In jeder Runde wird zuerst die rechte Blockhälfte als neue linke Blockhälfte für die nächste Runde abgespeichert. Der rechte Block wird mit Teilen des hinterlegten Passworts (Rundenschlüssel) mittels einer festgelegten Funktion verschlüsselt und mit dem linken Block aus der aktuellen Runde exklusiv-oder-verknüpft (XOR) und als neue rechte Blockhälfte für die nächste Runde abgespeichert. Die Verschlüsselung erfolgt in der Regel über 16 Runden. Danach werden beide Blöcke zum 64-bit Geheimtextblock zusammengefasst. Zur Entschlüsselung wird das oben beschriebene Verfahren in umgekehrter Reichenfolge ausgeführt.[74]

Der Data Encryption Standard (DES), der im Jahr 1977 vom NIST veröffentlicht wurde, ist der erste Verschlüsselungsalgorithmus, der auf dem Feistel-Netzwerk beruht. Der DES arbeitet mit einem 64-Bit-Schlüssel, wovon 8 Paritätsbits zur Fehlererkennung abgeschnitten werden und nur 56-Bit effektiv der Schlüssellänge

[73] Angelehnt an **[Sch02]** Kap. 1.4.2, S. 8
[74] Vgl. ebd. Kap. 1.4.2, S. 7-8

dienen. Der Algorithmus gilt heutzutage aufgrund des kleinen 56-Bit Schüsselraums (2^{56}) als unsicher.[75]

Der Triple Data Encryption Standard (3DES) bedient sich der Mehrfachanwendung des DES-Algorithmus. Durch die dreifache Anwendung wird eine Schlüsselstärke von 168-Bit erzielt. 3DES gilt als sicher, ist aber durch die Dreifachverschlüsselung sehr langsam.[76]

Der Blowfish-Algorithmus wurde im Jahr 1994 von Bruce Schneider entwickelt und besitzt eine variable Schlüssellänge zwischen 32- und 448-Bit. Für die Ver- und Entschlüsselung werden 18 Rundenschlüssel je 32-Bit gebildet. Die Funktion gleicht der des Feistel-Netzwerks und verschlüsselt die Daten in 16 Runden. Nach einer Rundenfunktion werden die Ergebnisse der beiden Hälften mit den restlichen zwei Rundenschlüsseln XOR-verknüpft. Der Blowfish-Algorithmus gilt heutzutage noch als sicher und verfügt über eine variable Schlüssellänge. Ein weiterer Vorteil ist auch die hohe Verarbeitungsgeschwindigkeit.[77]

Der CAST-Algorithmus wurde im Jahr 1996 veröffentlicht und ist nach den Anfangsbuchstaben seiner Entwickler Carlisle Adams und Stafford Tavares benannt. CAST zählt aktuell zu den effizientesten Verschlüsselungsalgorithmen basierend auf Feistel. Wie die meisten Blockchiffren besitz CAST eine feste Blocklänge von 64-Bit und eine flexible Schlüssellänge von 40- bis 128-Bit. Ein Klartextblock wird zuerst in zwei gleich große Hälften geteilt, und in 12 Runden (Schlüssellänge bis zu 80 Bit) bzw. 16 Runden (größer 80 Bit) verarbeitet. Dazu wird in jeder Runde ein Teil der Ausgabe mit dem Rundenschlüssel und den Operationen Addition, Subtraktion und XOR verknüpft. Diese Operationen (Addition, Subtraktion und XOR) werden in jeder Runde in unterschiedlicher Reihenfolge angewandt. Danach werden beide Blockhälften mit der XOR-Verknüpfung zusammengeführt. CAST arbeitet sehr schnell und gilt als sicherer Verschlüsselungsalgorithmus.[78]

Wegen der aufkommenden Unsicherheit im Bereich des DES Verfahrens wurde der Verschlüsselungsalgorithmus AES (Advanced Encryption Standard) entwickelt und im Jahr 2000 durch NIST bekanntgegeben. Der AES besitzt eine feste Blocklänge von 128-Bit und verwendet drei unterschiedliche Schlüssellängen (128-, 192- oder 256-Bit).[79] AES beruht nicht auf einem Feistel-Netzwerk, sondern aus einer zweidimensionalen Tabelle. Die Tabelle ist in vier Zeilen und Spalten aufgeteilt, wovon jede Zelle ein Byte groß ist. Die Transformation erfolgt in

[75] Vgl. **[Ern15]** Kap. 4.2.1, S. 150
[76] Vgl. ebd. Kap. 4.2.1, S. 151
[77] Vgl. **[Sch05]**
[78] Vgl. **[RFC2144]**
[79] Vgl. **[Ern15]** Kap. 4.2.2, S. 151-153

Runden, wobei bei jedem Durchlauf ein Teil des Originalschlüssels auf einen Klartextblock angewandt wird. Die Rundenanzahl ist abhängig von der verwendeten Schlüssellänge und beträgt 10 (AES-128), 12 (AES-192) oder 14 (AES-256) Runden. Neben der höheren Sicherheit arbeitet AES auch ungefähr dreimal schneller als DES.[80]

4.4.2 Stromverschlüsselung

Bei der Stromverschlüsselung erfolgt die Verschlüsselung bit-, byte- oder zeichenweise. Die Bit, Byte oder Zeichen des Klartextes werden durch einen Datenstrom von Schüsselfolgen einzeln XOR verknüpft. Der Klartext wird sofort im Gegensatz zur Blockverschlüsselung in einen Geheimtext übersetzt und ist daher für Echtzeitanwendungen geeignet.[81]

Der Ron's Code 4 (RC4) wurde im Jahr 1987 von Ronals L. Rivest für die Firma RSA Security Inc. entwickelt. Zur Schlüsselgenerierung wird ein Pseudo-Noise-Generator eingesetzt, der für jedes Klartextbyte ein Pseudo-Zufalls-Byte erstellt. Der Pseudo-Noise-Generator arbeitet mit Substitutionslisten, die 256-Byte groß sind und den Wertebereich 0-255 aufweisen. Der KSA-Algorithmus (Key-Scheduling Algorithm) vertauscht die Bytes in der Liste pseudo-zufällig anhand des RC4 Schüssels. Die Werte werden danach zufällig aus der Liste ausgelesen und als Pseudo-Zufalls-Byte ausgegeben. Der Geheimtext wird durch die bitweise Addition Modulo 2 des Klartextbyte und dem Pseudo-Zufalls-Byte generiert. RC4 verfügt über keinen Integritätsschutz und gilt seit dem Jahr 2013 als unsicher.[82]

4.4.3 Hashfunktion

Die Hashfunktion berechnet für beliebige Daten wie auch für Dateien eine Prüfsumme (Hashwert). Der Hash-Algorithmus bildet dabei eine große Eingabemenge auf einen fest definierten kleinen Hashwert ab. Eine Hashfunktion, die kollisionsresistent (unterschiedliche Hashwert für jede Datei) ist und die Einwegfunktion (nicht umkehrbar) beherrscht, wird als kryptographische Hashfunktion bezeichnet.[83] Die kryptographische Hashfunktion wird unter anderem bei der Übertragung von Passwörtern verwendet. Bei der Authentifizierung wird das Passwort im Klartext eingegeben und durch einen Algorithmus in einen Hashwert gewandelt, zum Zielsystem übertragen und mit dem dort hinterlegten Hashwert (z.B. in einer Datenbank) überprüft. Dadurch ist das Passwort des

[80] Vgl. **[Fis08]** S. 189
[81] Vgl. **[Spi11]** Kap. 1.3.2.2, S. 24-25
[82] Vgl. ebd. Kap. 2.4.1, S. 69-70
[83] Vgl. **[Sch06]** Kap. 2.1.2.3, S. 89

Benutzers nicht im Klartext ersichtlich und sogar theoretisch sicher, falls Cyberkriminelle durch eine Datenpanne an den Hashwert gelangt.[84]

Der Message-Digest Algorithm 5 (MD5) wurde 1991 von Roland L. Rivest entwickelt. MD5 erzeugt aus einem beliebigen Klartext einen 128-Bit-Hashwert (32 Hex-Zeichen).[85] Das Verfahren ist nicht kollisionsresistent, da es vorkommen kann, das zwei unterschiedliche Nachrichten denselben MD5-Hashwert ergeben.[86]

Der Secure Hash Algorithm (SHA) wurde von dem National Institute of Standards and Technology (NIST) und der National Security Agency (NSA) entwickelt und im Jahr 1993 veröffentlicht. SHA auch als SHA-0 bekannt, basiert auf Message-Digest Algorithm, kam aber wegen eines Konstruktionsfehlers nie zum Einsatz. SHA-1 wurde im Jahr 1995 veröffentlicht und erzeugt aus einem beliebigen Klartext einen 160-Bit-Hashwert (40 Hex-Zeichen). Auf Grund von bekannt gewordenen Angriffen auf SHA-1 verschlüsselte Daten wurde im Jahr 2005 der Einsatz von SHA-2 Hash-Algorithmen durch das NIST empfohlen. SHA-2 beinhaltet Hash-Funktionen mit vier unterschiedlichen Schlüssellängen (SHA-224, SHA-256, SHA-384 und SHA-512). SHA-2 gilt heutzutage noch als sicherer Verschlüsselungsstandard.[87]

4.4.4 Datenbankverschlüsselung

Abhängig von den Daten, Datenschutzrichtlinien und Anforderungen kann es notwendig sein, Daten innerhalb der Datenbank zu verschlüsseln.[88] Im Vordergrund der Datenbanksicherheit steht immer das Verhindern von Sicherheitsverletzungen, um die Schutzziele der Informationssicherheit zu wahren. Kommt es dennoch zu einem Datenverlust (Datenklau) ist die Verschlüsselung die letzte Instanz, da verschlüsselte Datensätze für die Diebe unbrauchbar sind. Die Datenbankverschlüsselung kann auf unterschiedliche Art implementiert werden.

Die Verschlüsselung auf Spaltenebene (Column-Level Encryption) ermöglicht die spaltenweise Verschlüsselung sensibler Daten (z.B. Kundendaten). Bei der Verschlüsselung werden die Dateninhalte einer bestimmten Spalte mit demselben Passwort und Verschlüsselungsalgorithmus verschlüsselt.[89]

Die Tabellenraum-Verschlüsselung (Tablespace Encryption) verschlüsselt alle Daten innerhalb eines referenzierten Tabellenraums. Für die erstmalige Verwendung muss eine neue Tabelle mit dem Zusatz „encryption“ erzeugt werden und ein Initialpasswort wie auch die Verschlüsselungsmethode (z.B. AES-128)

[84] Vgl. **[Spi11]** Kap. 3.1, S. 95-96
[85] Vgl. **[RFC1321]**
[86] Vgl. **[Pet]** S. 10-13
[87] Vgl. **[Tip05]** Kap. 5.2, S. 1351
[88] Vgl. **[BSI16]** Kap. M 4.72, S. 3286
[89] Vgl. **[IBM09]**

gewählt werden. Die Tabellenraum-Verschlüsselung biete eine größere Sicherheit, da alle Daten einer Tabelle verschlüsselt werden.[90]

Die Datenbankverschlüsselung (Database Encryption) auch Data-at-Rest-Encryption ist eine native Datenbankverschlüsselung, die den gesamten Inhalt der Datenbank (Tabellen und Logfiles) verschlüsselt. Die Datenbankschlüssel können auf einem externen Key-Server gespeichert werden. Die Datenbankverschlüsselung ist auf Grund der gesamtheitlichen Verschlüsselung mit einem geringen Konfigurations- und Pflegeaufwand betreibbar.[91]

4.5 Auditing

Da keine hundertprozentige Datenbanksicherheit existiert, ist die Nachvollziehbarkeit (Auditing) ein zentraler Aspekt, um Sicherheitsvorfälle von versuchten oder erfolgreichen Angriffen zu erkennen. Das Auditing kann auf unterschiedlichste Elemente eines Datenbanksystems angewandt werden: [92]

- SQL-Befehle wie die Data Manipulation Language (DML), die Data Definition Language (DDL) und die Data Control Language (DCL);
- Benutzer- und Administrator-Aktivitäten;
- Anmeldeversuche;
- Privilegien sowie Zugriffe auf Zeilen- und Spaltenebene.

Die Ereignisse werden mit der aktuellen Uhrzeit und Datum sowie dem ausführenden Benutzerprofil protokolliert (Monitoring). Diese Ereignisdaten werden entweder in einer speziellen Tabelle in der Datenbank selbst oder in einem vordefinierten Verzeichnis des Dateisystems als sogenannte Logdateien gespeichert.[93] Das Auditing von Datenbankaktivitäten kann auch zur Überprüfung von Anforderungen zur Einhaltung unternehmensinterner, gesetzlicher und vertraglicher Vorgaben (IT-Compliance) zum Einsatz kommen.[94]

4.6 Sicherheitsupdate

Um Datenbanksysteme vor Schwachstellen (bestehende wie auch neu bekannt gewordene) zu schützen, veröffentlichen die Softwarehersteller laufend Sicherheitsupdates.

4.6.1 Softwarepflege

Durch Softwareaktualisierungspakete werden Fehler in der Software behoben, die Softwarequalität verbessert oder der Funktionsumfang erweitert. Je nach Umfang,

[90] Vgl. **[Kyt10]** Kap. 16, S. 741
[91] Vgl. **[MarDE]**
[92] Vgl. **[Frö14]** Kap. 11.2, S. 273-274
[93] Vgl. ebd. Kap. 11.2, S. 274-275
[94] Vgl. **[Tie13]** Kap. 1, S.38

Inhalt und Wichtigkeit werden Softwareaktualisierungen in unterschiedlichen Paketen bereitgestellt. Der häufigste Grund von Aktualisierungen sind jedoch Fehlerkorrekturen, da sich Software nicht zwingend fehlerfrei entwickeln lässt. Softwareaktualisierungspakete unterscheiden sich nach Art und Umfang.

Das Software-Update enthält eine neue Version des Softwareprodukts (inkrementierte Versionsnummer) mit Softwareneuerungen und -verbesserungen wie auch mit Sicherheitsaktualisierungen der verschiedenen Komponenten und Bibliotheken.

Ein Maintenance Release ist eine Softwareaktualisierung, in der ausschließlich Fehler (Bugs) behoben werden.[95]

Das Service Pack umfasst eine größere Anzahl von Aktualisierungen, die viele offene Schwachstellen beheben, und alle Hotfixes, Updates und Maintenance Releases, die dem Service Pack zeitlich vorangehen, bereitstellt.[96]

Mit einem Security Patch wird ein Sicherheitsupdate, das einen bekannten Fehler in der betroffenen Software schließt und so die Ausnutzung der Sicherheitslücke unterbindet, bereitgestellt. Sicherheitsupdates sind die gängigste Methode, um bekannt gewordene Sicherheitslücken zu schließen. Sicherheitsupdates werden in der Regel an einem festgelegten Bereitstellungstag (Patchday) veröffentlicht.[97]

Der Hotfix enthält ein Sicherheitsupdate, das schnell und gezielt einen gravierenden Sicherheitsfehler kurzfristig behebt. Die Aktualisierung wird sofort nach Fertigstellung veröffentlicht und nur geringfügig getestet. Ein Hotfix sollte nur eingespielt werden, wenn die Software direkt von dem bekannten Fehler betroffen ist.[98]

4.6.2 Aktualisierungsprozess

Die verbundenen Ausfallzeiten (Downtimes) während der Einspielung von Sicherheitsupdates wie auch die Komplexität des Aktualisierungsprozesses sind ein weiterer Grund für schlecht gepflegte Datenbanksysteme. Der Aktualisierungsprozess wird je nach eingesetztem Verfahren unterschiedlich umgesetzt.

Die Aktualisierung beim In-Place Upgrade Verfahren umfasst das Herunterfahren des Datenbanksystems, die Aktualisierung (Austausch) der Binär- und Paketdateien sowie den Neustart des Datenbanksystems. Das In-Place Upgrade

[95] Vgl. **[MarRN]**
[96] Vgl. **[Tho11]** Kap. 3, S. 97
[97] Vgl. ebd. Kap. 3, S. 97
[98] Vgl. ebd. Kap. 3, S. 97

(Minor Upgrades) Verfahren kann nur auf direkt aufeinander folgende Release Level angewandt werden (z.B. Version 5.0 auf 5.1).[99]

Beim Major Upgrade Verfahren (z.B. Version 5.x auf 6.0) müssen vor der Aktualisierung alle Datenbanken sowie gespeicherte Prozeduren, Events und Konfigurationsdateien aus dem Datenbanksystem exportiert und gesichert werden. Danach erfolgt eine Neuinstallation des Datenbanksystems in der neuesten Version. Anschließend wird die neue Version gestartet und nach der Anmeldung mit einem administrativen Konto das zuvor gesicherte Abbild des Datenbanksystems in die neue Datenbankumgebung geladen. Danach muss noch, falls erforderlich, eine Upgrade-bedingte Migration, Erweiterung und Überprüfung aller relevanten Datenbankkomponenten erfolgen.[100]

4.7 Sicherheitsvorfälle

Mit dem Common Vulnerabilities and Exposures (CVE) Industriestandard wurde die Möglichkeit geschaffen, vorhandene Schwachstellen in Softwareprodukten über einen längeren Zeitraum zu dokumentieren und die Identifikation von Sicherheitslücken (Security Vulnerabilites) eindeutig zuzuordnen. Dafür wird der CVE-Präfix + Jahreszahl + beliebige Zahl verwendet. Die CVE-Kennung CVE-2015-4772 verweist z.B. auf eine Sicherheitslücke aus dem Jahr 2015 beim Oracle MySQL Server 5.6.24 und früheren Versionen.[101]

4.7.1 Common Vulnerability Scoring System

Das Common Vulnerability Scoring System (CVSS) ist ein offener Industriestandard zur Bewertung des Schweregrads einer Sicherheitslücke.[102] Der Schweregrad berechnet sich aus den drei Faktoren: Base Metriken (Bewertung der Schwachstelle), Temporal Metriken (Besonderheiten der Sicherheitslücke zum Zeitpunkt der Untersuchung) und Environmental Metriken (umgebungsspezifischen Faktoren) und wird mit dem Wert zwischen 0 bis 10 angegeben.[103] Geläufiger ist die Einteilung des Schweregrads in die textliche Bewertung nach dem CVSS v3.0 aus dem Jahr 2015:[104]

Kritisch (9.0-10.0): Schwachstelle, die Codeausführung ohne Zutun des Benutzers hervorruft und sich so selbst verbreitet (z.B. Würmer).
Kriterien: Einschleusen und Ausführen von Code; Fernzugriff

[99] Vgl. **[MyS15]** Kap. 2.10.1.3, S. 215-216
[100] Vgl. ebd. Kap. 2.10.1.3, S. 215-216
[101] Vgl. **[CVE]**
[102] Vgl. **[Mel07]**
[103] Vgl. **[Röc07]**
[104] Vgl. **[CVSS]** S. 16

(Remote Access); Exploit-Code ist öffentlich zugänglich und die Schwachstelle kann aktiv ausgenutzt werden.[105]

Hoch (7,0-8,9): Schwachstelle, die die Vertraulichkeit, die Integrität oder die Verfügbarkeit von sensitiven Daten oder anderen Ressourcen gefährdet.
Kriterien: Einschleusen und Ausführen von Code; Fernzugriff; keine Exploits oder Angriffe bekannt[106]

Mittel (4,0-6,9): Schwachstelle ist z.B. auf die Authentifizierung oder die Standardkonfiguration beschränkt.
Kriterien: Datenverlust; Denial-of-Service; Fernzugriff[107]

Niedrig (0,1-3,9): Schwachstelle wirkt sich nur auf spezielle Eigenschaften der betroffenen Software aus.
Kriterien: lokale Rechteausweitung oder Datenverlust möglich[108]

Keine (0): keine Schwachstelle bekannt

4.7.2 Bewertung

Für die Bewertung der Sicherheitsvorfälle wurde die US-amerikanischen National Vulnerability Database (NVD) der Bundesbehörde NIST genutzt. Die frei zugängliche Schwachstellen-Datenbank ist ein digitales Verzeichnis, das die Bewertung der Datenbanksicherheit nach der Anzahl der Sicherheitsvorfälle und des Schweregrads der Schwachstelle ermöglicht. Abweichend zu dem in Kapitel 4.7.1 vorgestellten Bewertungssystem CVSS v3.0 wird für den Vergleich das alte Bewertungssystem CVSS v2.0 mit der Einstufung: niedrig (0-3), mittel (4-6) und hoch (7-10) verwendet. Grund dafür ist, dass die bekannt gewordenen Schwachstellen der letzten Jahre noch nicht in das neue Bewertungssystem CVSS v3.0 überführt wurden.[109]

[105] Vgl. **[Mic11]**
[106] Vgl. ebd.
[107] Vgl. ebd.
[108] Vgl. ebd.
[109] Vgl. **[NVD]**

5 Relationale Datenbanksysteme

In Kapitel 5 werden die allgemein beschriebenen Sicherheitsaspekte aus Kapitel 4 auf Basis der Implementierungen von den Open-Source-Datenbanksystemen MySQL Community Server 5.7.10, MariaDB 10.1.11 und PostgreSQL 9.5.1 sowie der kommerziellen Oracle Database 12c Enterprise Edition aufgezeigt.

5.1 MySQL

MySQL zählt mit ungefähr 50 Millionen Installationen zu den beliebtesten relationalen Datenbanksystemen weltweit. Viele Content Management Systeme (CMS) und eine Vielzahl von anderen Webanwendungen nutzen MySQL zur Datenspeicherung.[110] MySQL gehört zwischenzeitlich zu Oracle und ist unter der GPL-Lizenz (General Public License) für freie Software erhältlich. Neben dem freien MySQL Community Server vertreibt Oracle auch die kommerzielle MySQL Enterprise Edition. MySQL wurde 1994 von Michael Widenius zusammen mit David Axmark und Allan Larsson im schwedischen Softwareunternehmen MySQL AB entwickelt. Das Datenbanksystem wurde als proprietäre Software im Jahre 1995 veröffentlicht. Ab dem Jahre 2000 wurde MySQL in der Version 3.23.19 in die General Public License für freie Software aufgenommen. Im Februar 2008 wurde das MySQL-Datenbanksystem von dem Unternehmen Sun Microsystems aufgekauft. Im Januar 2010 hat die Oracle Corporation Sun Microsystems gekauft.[111] Im Dezember 2012 gründeten die ehemaligen MySQL-Entwickler Michael Widenius, David Axmark und Allan Larsson die unabhängige MariaDB-Foundation. MariaDB ist eine quelloffene Abspaltung (Fork) von MySQL. MySQL liegt aktuell in der Version 5.7 vor.[112]

5.1.1 Konfiguration

Das MySQL-Datenbanksystem ist für fast alle Betriebssysteme und Plattformen (z.B. Apple Mac OS, diverse Linux Betriebssysteme, Microsoft, IBM System iOS und AIX, HP UX, Sun Solaris sowie Novell Netware) erhältlich. Bei Microsoft Windows erfolgt die Installation nach dem Herunterladen (Download) der Installationsdatei über die grafische Benutzeroberfläche MySQL Installer GUI. Alternativ kann MySQL aber auch per Eingabeaufforderung mittels des MySQLInstallerConsole installiert werden.

Die grafische Benutzeroberfläche bei Windows-Betriebssystemen führt den Benutzer durch die Installationsroutine. Während der Installation des Datenbanksystems wird im ersten Konfigurationsschritt (Type and Networking) die Verwendungsart z.B. Development-, Server- oder Dedicted Machine gewählt,

[110] Vgl. **[Sch13]**
[111] Vgl. **[Mey13]**
[112] Vgl. **[MarOS]**

wodurch die maximale Auslastung der Hardwareressourcen (z.B. des Arbeitsspeichers) definiert wird. Im gleichen Schritt kann der externe Zugriff, der standardmäßig über den MySQL-Dienst auf Port 3306 lauscht, geändert oder deaktiviert werden. Der zweite Konfigurationspunkt (Accounts and Roles) verlangt die Eingabe des MySQL-Root-Passworts und gibt gleichzeitig Auskunft über die Passwortstärke. Weiter lassen sich Datenbankbenutzer mit festgelegten Rollen (z.B. Backup, Designer, Security) einrichten. Im dritten Schritt (Windows Service) wird der MySQL-Server als Systemdienst konfiguriert, der standardmäßig als Hintergrundprozess läuft. Die Konfiguration ermöglicht die Auswahl des Systemdienstnamens, die Autostartfunktion sowie das Benutzerkonto (Account) unter dem der MySQL-Systemdienst später läuft. Zur Auswahl steht der Standard System Account oder der Custom User in dem der MySQL-Systemdienst unter Windows betrieben werden kann. Ein wichtiger Sicherheitsaspekt ist das der MySQL-Systemdienst mit den möglichst geringsten Berechtigungen am Betriebssystem betrieben wird, um potentielle Gefahren bei einem erfolgreichen Angriff zu minimieren. Durch weitreichende Berechtigungen (z.B. Administrator) erlangen die Angreifer kompletten Zugriff auf das Betriebssystem und die Dateisystemebene (z.B. Logdateien, Data Dumps). Der letzte Konfigurationsschritt (Advanced Options) ermöglicht es Logdateien zur Kontrolle und Überwachung (Auditing) zu aktiveren.

Unter Linux wird MySQL direkt aus den Paketquellen installiert. Dazu wird z.B. unter Debian der Konsolenbefehl apt-get install mysql-server oder im grafischen Paketmanager das Paket des MySQL-Server ausgewählt. Die Installationsroutine unter Linux verlangt während der Installation lediglich die Eingabe eines Root-Passwort zur Absicherung des Datenbanksystems. Standardmäßig ist bei allen Installationsarten von MySQL keine Protokollierung aktiv.[113]

Die Konfigurationsdatei „my.cnf“ unter Linux oder „my.ini“ bei Microsoft Windows (siehe Abbildung 4) dient der erweiterten Konfiguration und Anpassung des Datenbanksystems.

[113] Vgl. **[MyS15]** Kap. 2.3.3.1, S. 68-93

```
[client]
no-beep
port=3306
[mysql]
default-character-set=utf8
[mysqld]
port=3306
datadir=C:/ProgramData/MySQL/MySQL Server 5.7/Data
character-set-server=utf8
default-storage-engine=INNODB
sql-mode="STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"
log-output=FILE
general-log=1
general_log_file="WInf.log"
slow-query-log=1
slow_query_log_file="WInf-slow.log"
long_query_time=10
log-bin="WInf-bin"
log-error="WInf.err"
```

Abbildung 4 - MySQL Konfigurationsdatei

Um die Informationssicherheit zu gewährleisten, muss die Standardkonfiguration noch um wichtige Sicherheitsaspekte wie z.B. TLS erweitert werden. Hierzu bietet MySQL neben dem manuellen Editieren das Werkzeug mysql_secure_installation an. Seit der MySQL Version 5.7.4 wurden die Sicherheitsrichtlinien von MySQL zusätzlich verschärft. Bei jeder Neuinstallation wird nur noch ein administratives Konto root@localhost mit einem zufällig erzeugten Passwort erzeugt. Dieses Passwort ist als „expired“ gekennzeichnet und muss bei erster Benutzung geändert werden. Ein anonymer Benutzer wird nicht mehr bereitgestellt; auf die Testdatenbank „Test“ wird komplett verzichtet.[114]

5.1.2 Authentifizierung

In der Standardkonfiguration verwendet MySQL zur Authentifizierung von Clients am Datenbanksystem die Benutzername-Passwort-Authentifizierung. Die Anmeldung am Datenbanksystem wird durch den Client initialisiert. Dazu wird der Hostname des Datenbankservers, der Benutzername, das Passwort und die Datenbank (z.B. mysql --host=localhost --user=myname --password=mypass mydb) zur Authentifizierung an den DB-Sever gesendet. Die Anmeldeinformationen der Benutzerkonten wie die Hostnamen oder IP-Adressen der Clientsysteme sind in der DB-Tabelle „user“ der Systemdatenbank „mysql“ abgelegt.

[114] Vgl. **[Ora14]**

```
mysql> Select user, authentication_string, host, plugin from mysql.user;
+------------+-------------------------------------------------------------------+-----------+-----------------------+
| user       | authentication_string                                             | host      | plugin                |
+------------+-------------------------------------------------------------------+-----------+-----------------------+
| root       | *00A51F3F48415C7D4E8908980D443C29C69B60C9                          | localhost | mysql_native_password |
| mysql.sys  | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE                          | localhost | mysql_native_password |
| sha256user | $5$/b_▯-%KX▯Hn%GrQ_qB▯~$Bzw600zgCBS2p0ZsvPzr5N3UZwD7lKNYOLvwcqRsvC6 | localhost | sha256_password       |
+------------+-------------------------------------------------------------------+-----------+-----------------------+
3 rows in set (0.00 sec)
```

Abbildung 5 - MySQL Authentifizierung

In der Standardkonfiguration ist die native Passwort Authentifizierungsmethode (mysql_native_password) zur Authentifizierung aktiv.[115] Beim Anlegen der Benutzerkonten wird mit dem Secure Hash Algorithm das Benutzerpasswort in eine 41-stellige Hexadezimalzahl verschlüsselt und in der Tabelle „mysql.user“ abgespeichert.[116] Beim Anmeldeprozess wird das im Klartext übermittelte Passwort mit der SHA-1 Hashfunktion in eine 41-stellige Hexadezimalzahl gewandelt und mit dem Wert in der Systemtabelle „mysql.user“ verglichen (siehe Abbildung 5).[117] Falls dem Benutzerkonto eine Erweiterung zur Authentifizierung (z.B. sha256_password) hinterlegt ist, ruft der MySQL-Server die Erweiterung (Plugin) auf, um den Benutzer über das hinterlegte Authentifizierungsprotokoll zu authentifizieren.

Die SHA-256 Authentifizierung (sha256_password) beruht auf der Benutzername-Passwort-Authentifizierung und verwendet zur Verschlüsselung des Passwortes den SHA-2 Hash-Algorithmus mit einer 256-Bit Schlüssellänge. Die Kommunikationsverbindung kann nur ausschließlich über eine SSL/TLS-Verbindung aufgebaut werden, die in der Standardkonfiguration nicht aktiviert ist und nachträglich konfiguriert werden muss.[118]

Die Erweiterung ohne Authentifizierung (mysql_no_login) ermöglicht den Zugriff auf das Datenbanksystem ohne Überprüfung der persönlichen Identität. Alle Clients die sich verbinden möchten, verwenden die identische Benutzerkennung und können dadurch vordefinierte Abfragen an das Datenbanksystem stellen, um begrenzte Informationen zu erhalten.[119]

Die Authentifizierung per Socket (auth_socket) ermöglicht die lokale Authentifizierung bei Linux-Betriebssystemen durch die Unix-Socket Datei. Der lokale Benutzername des Betriebssystems (Socket User) muss mit dem

[115] Vgl. **[MyS15]** Kap. 6.3.8, S. 915-917
[116] Vgl. ebd. Kap. 6.1.2.4, S. 862-863
[117] Vgl. **[Far15]**
[118] Vgl. **[MyS15]** Kap. 6.3.9.4, S. 924-927
[119] Vgl. ebd. Kap. 6.3.9.5, S. 927-928

authentication_string der in der „mysql.user“ Systemtabelle hinterlegt ist übereinstimmen.[120]

Der MySQL Community Server bietet zur Authentifizierung abgesehen von der lokalen Socket Authentifizierung nur die Benutzername-Passwort-Authentifizierung an. In der Standardkonfiguration sind die native und die SHA-256 Authentifizierungsmethode aktiv. Beide Verfahren verfügen über einen Häufigkeitsschutz (Limitations) und automatische Passworterneuerung (Password Expiration Policy). Die native Methode speichert das Passwort in dem veralteten und unsicheren SHA-1 Hash-Algorithmus. Weiterhin wird bei der Authentifizierung das Passwort ohne aktivierte Übertragungssicherheit (SSL/TLS) im Klartext übertragen. Die Sicherheit der Passwörter beider Methoden hängt einzig von der „Unbekanntheit“ des Hashwertes ab. Gelingt einem Angreifer der Zugriff auf die „mysql.user“ Systemtabelle oder das Einsehen der Logdateien die eventuell den Hashwert enthalten, besteht die Möglichkeit, dass der Angreifer an das Passwort im Klartext gelangt. Die Passwortsicherheit kann durch lange Passwörter (maximal 32 Zeichen bei MySQL) die aus einer Kombination aus großen und kleinen Zeichen, Zahlen oder Sonderzeichen bestehen, vor unautorisierten Zugriff sowie vor der ungewollten Entschlüsselung geschützt werden.

5.1.3 Zugriffskontrolle

Die Zugriffkontrolle des MySQL-Datenbanksystem basiert auf der Discretionary Access Control. MySQL selbst bezeichnet die benutzerbestimmbare Zugriffskontrolle intern als Access Privilege System. Das Access Privilege System steuert, welche Berechtigungen die Benutzer am Datenbanksystem zugewiesen bekommen. Die Zugriffsrechte sind dabei eindeutig an die Identität der einzelnen Benutzer geknüpft. Die Einmaligkeit der Identität wird durch das Authentifizierungsverfahren anhand der Hostadresse des Clientsystems, dem Benutzernamen und dem Passwort sichergestellt. Nach der erfolgreichen Authentifizierung werden die Berechtigungen anhand der Identität zugewiesen und durch die Zugriffskontrolle gewahrt. Dazu wird bei jeder Anfrage (z.B. Werte aus einer Tabelle auslesen) an das Datenbanksystem geprüft ob der Benutzer mit der besagten Identität über ausreichende Berechtigungen (z.B. SELECT-Befehl) zur Ausführung besitzt.[121]

Die Informationen der Zugriffsberechtigung sind in den Grant-Tabellen (user, db, tables_priv, columns_priv und procs_priv) der „mysql“ Datenbank hinterlegt. Die Informationen der benutzerbestimmbaren Zugriffsberechtigung werden bei jedem Systemstart aus den Grant-Tabellen in den Speicher (In-Memory-Tables)

[120] Vgl. **[MyS15]** Kap. 6.3.9.7, S. 929-930
[121] Vgl. ebd. Kap. 6.2, S. 879-880

geladen.[122] Bei einer Änderung der Zugriffsberechtigung werden die Grant-Tabellen im Speicher, je nach Befehl (z.B. GRANT oder INSERT) direkt oder erst nach einem Neustart des Datenbanksystems in den Speicher geladen.[123]

	Name der Grant-Tabelle in MySQL				
	user	db	tables_priv	columns_priv	procs_priv
Scope columns	Host User	Host Db User	Host Db User	Host Db User	Host Db User
Privilege columns	Select_priv usw.	Select_priv usw.	Table_name	Table_name Column_name	Routine_name Routine_type
Security columns	authentication _string plugin				
Resource control columns	max_connect- ions				

Tabelle 2 - MySQL Zugriffssteuerungsliste[124]

In Tabelle 2 sind die Grant-Tabellen dargestellt (SQL-Befehl: DESCRIBE USER;), die zur Realisierung der Discretionary Access Control in MySQL benötigt werden. Alle Grant-Tabellen enthalten einen Geltungsbereich (Scope columns) der sich aus der Hostadresse (Host), der Datenbank (DB) und dem Benutzer (User) zusammensetzt, sowie den Berechtigungsfeldern (Privilege columns) die beschreiben welche Handlungen ein Benutzer im vorgegebenen Geltungsbereich ausführen darf. Die Grant-Tabelle „user“ dient der eindeutigen Authentifizierung von Benutzern anhand der Hostadresse des Clientsystems (Host), des Benutzernamens (User) und dem Passwort (authentication_string). Die Grant-Tabelle „db“ bestimmt anhand des Geltungsbereichs auf welche Datenbank ein angemeldeter Benutzer Zugriffrechte erlangt. Die Grant-Tabelle „tables_priv“ definiert auf welche Tabellen ein Benutzer innerhalb der Datenbank zugreifen darf. Die Grant-Tabelle „columns_priv“ hingegen grenzt die Spalten innerhalb der Tabelle ab. Weiter enthält die Grant-Tabelle „procs_priv“ die Routinen die ein Benutzer zur Ausführung bringen darf.[125]

5.1.4 Verschlüsselung

Sensitive Dateninhalte lassen sich in MySQL auf Spaltenebene (Column-Level Encryption) verschlüsseln. Durch Einsatz eines vorgegebenen Verschlüsselungsverfahren werden die Inhalte jedes einzelnen Elements der zu verschlüsselnden Spalte/Spalten in der Datenbank verschlüsselt. In MySQL wird nur der offizielle AES-Algorithmus zur Verschlüsselung von Daten unterstützt.

[122] Vgl. **[MyS15]** Kap. 6.2.1, S. 880-881
[123] Vgl. ebd. Kap. 6.2.6, S. 896
[124] Angelehnt an ebd. Kap. 6.2.2, S. 885-891
[125] Vgl. ebd. Kap. 12.1, S. 329-330

Standardmäßig nutzt MySQL die 128-Bit-Schlüssellänge des AES-Standards. Bei entsprechender Konfiguration können auch die Schlüssellängen 196- oder 256-Bit verwendet werden.[126] Neben der Schlüssellänge ist es auch möglich, die Betriebsart (Mode Of Operation) der Blockverschlüsselung zu wählen. Der MySQL Community Version (kompiliert mit yaSSL) stehen standardmäßig der Electronic Code Book Mode (ECB Mode) und erweitert der Cipher Block Chaining Mode (CBC Mode) als Blockchiffrierung zu Verfügung.[127] ECB spiegelt den einfachsten Betriebsmodus der Blockverschlüsselung wieder. Jeder Klartextblock wird unabhängig der anderen Blöcke verschlüsselt. Dadurch enthalten zwei identische Klartextblöcke den gleichen Geheimblock. Im CBC Betriebsmodus wird zusätzlich zu jedem Klartextblock der zuvor erzeugte Geheimblock XOR-verknüpft, wodurch die Geheimhaltung der sensitiven Daten erhöht wird, da aus identischen Klartextblöcken jetzt unterschiedliche Geheimblöcke berechnet werden.[128]

5.1.5 Auditing

Zur Protokollierung von Benutzeraktivitäten bietet der MySQL Community Server nur die Ereignisprotokollierung (Logging) an. Zu jeder Datenbankaktivität wird zur späteren Nachvollziehbarkeit ein Eintrag der durchgeführten Aktion mit Zeitstempel in die jeweilige Logdatei geschrieben. Logdateien enthalten somit den chronologischen Ablauf von Nutzung und Veränderung der Datenbank. In der Standardkonfiguration sind alle Ereignisprotokolle in MySQL deaktiviert, außer dem Fehlerprotokoll (Error Log) bei Microsoft Windows. Das MySQL-Datenbanksystem enthält verschiedene Ereignisprotokolle zur Analyse der Benutzeraktivitäten (siehe Tabelle 3).[129]

Logtypen	Informationen die in die Logdatei geschrieben werden
Error log	Protokollierung beim Starten, Stoppen und Fehler im Betrieb
General query log	Verbindungsaubau von Clients und SQL-Statements
Binary log	Datenveränderungen
Relay log	Datenveränderungen veranlasst durch einen Master-Server
Slow query log	Anfragen mit Zeitüberschreitung
DDL log	Metadaten der Datenbanksprache (DDL)

Tabelle 3 - MySQL Logdateien[130]

Der Error Log enthält Informationen über Start bzw. Stopp des Datenbankdienstes sowie über kritische Fehler während der Laufzeit des MySQL-Servers (mysqld).

[126] Vgl. **[MyS15]** Kap. 12.13, S. 1481
[127] Vgl. ebd. Kap. 5.1.4, S. 597
[128] Vgl. **[Sch02]** Kap. 1.4.2, S. 8-9
[129] Vgl. **[MyS15]** Kap. 5.2.1, S. 809-811
[130] Angelehnt an ebd. Kap. 5.2, S. 809

Zusätzlich werden Hinweise in die Error Log geschrieben, wenn eine Tabelle überprüft oder repariert wurde.[131]

In der General Query Log werden alle Ereignisse protokolliert, die durch Clients ausgelöst werden: den Verbindungsaufbau/-abbau von Clients sowie alle SQL-Anweisungen, die von Clients an das Datenbanksystem gestellt werden.[132] Die General Query Log und Slow Query Log, können auch innerhalb der Datenbank in Tabellen gespeichert werden.

Die Binary Log enthält Einträge über Datenbankveränderungen (Modify Data) wie z.B. das Erstellen einer Tabelle oder das Ändern der Inhalte innerhalb der Tabelle. Lesende (nicht modifizierende) Zugriffe auf die Daten (Not Modify Data) wie z.B. beim SELECT und beim SHOW Kommando werden in der Binary Log nicht festgehalten. Jeder Log-Eintrag enthält eine Ausgabe, wie viel Zeit der Datenbankserver für die Kommandoausführung benötigt hat. Des Weiteren wird die Binary Log auch bei der Synchronisation der Datenspiegelung in einer Master-Slave-Replikation wie auch zur Datenwiederherstellung nach einem Datenbankcrash verwendet.[133]

Die Relay Log wird nur für Master-Slave-Replikationen benötigt, um Änderungen am Master-Server festzuhalten und später an der Slave-Replikation anzuwenden.[134] Die Relay Log besteht wie die Binary Log aus mehreren fortlaufend nummerierten Dateien, die Ereignisse speichern, sowie einer Index-Datei zur Verwaltung aller Relay Log Dateien.[135]

Im Slow Query Log werden alle SQL-Statements, die länger als die vorgeschriebene Zeit (long_query_time) benötigen, aufgeführt. In der Standardkonfiguration werden keine administrativen Statements sowie Anfragen, die keinen Index verwenden, protokolliert.[136]

Die DDL Log protokolliert Informationen, die von der Datenbanksprache DDL erzeugt werden (z.B. DROP TABLE und ALTER TABLE). Die Logdatei wird systembedingt benötigt, um Datenbankoperationen, die durch einen Absturz nicht vollständig oder korrekt, umgesetzt wurden, wiederherzustellen. Bei der DDL Log handelt es sich um eine Binärdatei, die nicht konfigurierbar und einsehbar ist.[137]

[131] Vgl. **[MyS15]** Kap. 5.2.2, S. 811-813
[132] Vgl. ebd. Kap. 5.2.3, S. 814-815
[133] Vgl. ebd. Kap. 5.2.4, S. 815-818
[134] Vgl. ebd. Kap. 5.2, S. 809
[135] Vgl. ebd. Kap. 17.2.4.1, S.2563-2564
[136] Vgl. ebd. Kap. 5.2.5, S. 827-828
[137] Vgl. ebd. Kap. 5.2.6, S. 828-829

5.1.6 Sicherheitsupdate

Sicherheitsupdates erscheinen bei MySQL im Turnus von drei Monaten. Die Sicherheitsupdates enthalten mehrere Fehlerbehebungen, die produktübergreifend an festgelegten Bereitstellungstagen in vierteljährlichen Abständen (z.B. am 18. Oktober 2016) erscheinen. Security Alerts auch bekannt als Hotfixes werden unregelmäßig veröffentlicht und beheben Sicherheitslücken, bei denen es zu kritisch wäre, bis auf den nächsten Bereitstellungstag zu warten.[138] Sicherheitsupdates müssen beim MySQL Community Server manuell eingespielt werden. Ausnahme bildet der Paketmanager unter Linux der Hotfixes automatisch erneuert. Das In-Place Upgrade Verfahren unterstützt Erneuerungen auf Serienversionen (Release Series Version) z.B. von Version 5.7.9 auf 5.7.10 oder auch Version 5.7.2 auf 5.7.10. Die Erneuerungen auf Veröffentlichungsebene (Release Level) müssen nacheinander erfolgen; ein Überspringen einer Zwischenversion ist nicht möglich (direkter Wechsel z.B. von Version 5.5 auf 5.7 geht nicht). Nach einer erfolgreichen Produktaktualisierung muss das Hilfsprogramm mysql_upgrade ausgeführt werden, um alle Tabellen einer Datenbank auf Inkompatibilitäten mit der neusten Produktversion zu prüfen. Weiterhin werden durch das Hilfsprogramm die Systemtabellen erneuert und bei Bedarf Berechtigungen und Funktionen der neusten Produktversion hinzugefügt.[139]

5.1.7 Sicherheitsvorfälle

Die Anzahl der MySQL-Sicherheitsvorfälle (siehe Tabelle 4) wurden durch eine Auswertung der NVD-Daten der letzten 5 Jahre (Auswertezeitraum 01. Januar 2011 bis 31. Dezember 2015) auf Basis des Stichworts „Oracle MySQL“ ermittelt.

Jahr	Niedrig (0-3)	Mittel (4-6)	Hoch (7-10)	Gesamt
2011	0	8	0	8
2012	13	45	4	62
2013	14	52	2	68
2014	18	43	4	65
2015	32	43	3	78
Gesamt	77	191	13	281

Tabelle 4 - MySQL Sicherheitsvorfälle

In dem ausgewerteten Zeitraum wurden insgesamt 281 Schwachstellen des MySQL-Datenbanksystems registriert. Als häufigste Schwachstelle mit 180 Sicherheitsvorfällen wurde „Unspecified vulnerability in Oracle MySQL-Server and earlier version, allows remote authenticated users to affect confidentiality/integrity/availability“ vermerkt. Diese Schwachstelle wird mit dem Schweregrad „Niedrig bis Mittel“ eingestuft, da zur Ausnutzung dieser

[138] Vgl. **[OraCP]**
[139] Vgl. **[Ora]**

Schwachstelle der Angreifer sich zuerst am Datenbanksystem authentifizieren muss. Die Schwachstelle ermöglicht die Einschränkung der Vertraulichkeit, der Integrität und der Verfügbarkeit.

5.2 MariaDB

MariaDB ist auf Basis einer Abspaltung (Fork) von MySQL entstanden und ist eine Open-Source-Alternative zum MySQL-Datenbanksystem. Neben der frei zugänglichen Community Version wird MariaDB auch als Enterprise Version angeboten. Die Enterprise Version ist speziell für Geschäftskunden zugeschnitten und beinhaltet einen Rundum-Support sowie einen erweiterten Funktions- und Sicherheitsumfang. MariaDB liegt aktuell in der Version 10.1.11 vor und stammt ursprünglich von den MySQL Versionen 5.5 ab.[140] Eine bestehende MySQL-Datenbank lässt sich auf einfache Weise in das MariaDB-Datenbanksystem importieren.

5.2.1 Konfiguration

Die grafische Installationsroutine ähnelt der von MySQL. Nach dem Herunterladen und Start der Installation kann ein Root-Passwort vergeben und die Zugriffsmethode „Lokal“ oder „Remote“ gewählt werden. Im nächsten Schritt lässt sich der Dienstname (mysql) und der Port, der standardmäßig auf der Netzwerk-Adresse 3306 lauscht, ändern. Weitergehende Konfigurationsschritte sind über die grafische Oberfläche nicht möglich. Zur Einrichtung stellt MariaDB dieselbe Konfigurationsprozedur wie MySQL zur Verfügung sowie die identische Konfigurationsdatei (my.cnf). MariaDB verfügt über ein zusätzliches grafisches Administrationswerkzeug namens „HeidiSQL“, das die Benutzer- und die Datenbankenverwaltung erleichtert.

5.2.2 Authentifizierung

Dem Datenbanksystem MariaDB steht wie MySQL die Standardauthentifizierung mysql_native_password und mysql_old_password zur Verfügung. Genauso wie in MySQL ist die lokale Socket Authentifizierung (unix_socket) per Erweiterung nutzbar.[141] Abweichend zu MySQL besteht die Möglichkeit die PAM-Erweiterung (auth_pam), eine Authentifizierung unter UNIX-artigen Betriebssystemen (z.B. Linux, FreeBSD, Solaris), zu verwenden. Mittels PAM ist es möglich, die verschiedensten Authentifizierungsverfahren wie das normale PAM-Verfahren, das LDAP, das Netzwerkprotokoll Secure Shell, Einweg Passwörter (One-Time Password) durch SMS-Nachrichten, die Zwei-Faktor-Authentifizierung (Two-Step Verification) oder die Kombinationen aus mehreren Authentifizierungsmethoden zu

[140] Vgl. **[MarK]**
[141] Vgl. **[MarUS]**

realisieren.[142] Speziell für Windows steht die named_pipe Erweiterung zur Verfügung. Diese Erweiterung ermöglicht es, die Anmeldeinformationen des Windows-Betriebssystems für die Datenbankauthentifizierung zu verwenden.[143] Weiter steht das Generic Security Service Application Program Interface (GSSAPI) eine Programmierschnittstelle für die namenlose Authentifizierung für Linux-Betriebssysteme bereit. Die Passwortsicherheit kann durch die Erweiterungen simple_password (stärkere Minimalvorgaben für die Wahl von Passwörtern wie z.B. Passwortlänge, Anzahl von Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen) und cracklib_password_check (Passwortvalidierung) weiter ausgebaut werden.[144]

5.2.3 Zugriffskontrolle

Anderes als MySQL, das der benutzerbestimmbaren Zugriffskontrolle (DAC) unterliegt, setzt das Datenbanksystem MariaDB auf die rollenbasierte Zugriffssteuerung (RBAC). Durch die rollenbasierte Zugriffskontrolle ist es möglich, große Benutzerzahlen mit geringem Administrationsaufwand zu verwalten. Dabei werden die Benutzer in spezielle Rollen bzw. Gruppen unterteilt. Bei großen Installationen sind in der Regel die Anzahl der benötigten Benutzerberechtigungen und Berechtigungsstufen viel geringer als die Anzahl der Benutzer. Somit wird zur Arbeitserleichterung zuerst die Rollen/Gruppen mit den erforderlichen Berechtigungen definiert und danach die Rollen den Benutzern zugewiesen.[145]

5.2.4 Verschlüsselung

Personenbezogene Daten, die dem Schutzbedarf unterliegen, können in MariaDB durch Verschlüsselung einer einzelnen Tabelle (PAGE_ENCRYPTION=1) oder der gesamten Datenbank inklusive der Logdateien erfolgen.[146] Die native Datenbankverschlüsselung, die ursprünglich von Google entwickelt wurde, speichert alle ruhenden Daten verschlüsselt (Data-at-Rest-Encryption) physikalisch auf dem Speichermedium und schützt so die Daten vor unberechtigtem Zugriff durch ständig rotierende Schlüssel.[147] MariaDB nutzt wie MySQL den AES-Algorithmus; alternativ kann bei MariaDB auch der DES-Algorithmus genutzt werden.[148]

[142] Vgl. **[Mar]**
[143] Vgl. **[Mar16]**
[144] Vgl. **[MarPV]**
[145] Vgl. **[MarR]**
[146] Vgl. **[MarTE]**
[147] Vgl. **[MarDE]**
[148] Vgl. **[MarEH]**

5.2.5 Auditing

MariaDB nutzt in gleicherweise wie MySQL die Ereignisprotokolle Error Log, General Query Log, Slow Query Log, Binary Log und Relay Log.[149] In der Standardkonfiguration ist nur die Error Log aktiviert (siehe Kapitel 5.1.5). Hinzu kann mit dem MariaDB Audit Plugin die Überwachung der Datenbank-Aktivitäten erweitert werden. Das Audit Plugin ist seit der Version 10.0.10 standardmäßig enthalten.

Logtypen	Informationen, die in die Logdatei geschrieben werden
CONNNECT	An- und Abmeldungen sowie fehlgeschlagene Anmeldungen inklusive der Fehlercodes
QUERY	Alle gestellten Abfragen und ihre Ergebnisse sowie Syntaxfehler und Berechtigungsverweigerungen
TABLE	Name der Tabelle, die von der Kommandoausführung betroffen war
QUERY_DDL	Abfragen die Data Definition Language (z.B. CREATE, ALTER etc.) enthalten
QUERY_DML	Abfragen die Data Manipulation Language (z.B. INSERT, UPDATE etc.) enthalten
QUERY_DCL	Abfragen die Data Control Language (z.B. GRANT, REVOKE etc.) enthalten

Tabelle 5 - MariaDB Auditing[150]

Um die erweiterten Auditfunktionen (siehe Tabelle 5) nutzen zu können, muss die Protokollierung einmalig durch den Aufruf SET GLOBAL server_audit_logging=on; oder in der Konfigurationsdatei server_audit_logging=on aktiviert werden. Alle Ereignisse (Logtypen) werden standardmäßig in einer zentralen Logdatei oder alternativ an den Ereignisprotokolldienst Syslog übergeben.[151] Weiter kann konfiguriert werden, wo die Logdatei hingeschrieben wird, ab welcher Dateigröße eine neue Logdatei erzeugt wird oder nach wie vielen Rotationen die Logdatei wieder überschrieben wird.[152]

5.2.6 Sicherheitsupdate

Sicherheitsupdates werden bei MariaDB zeitnah bereitgestellt. MariaDB verfolgt auch eine andere Sicherheitspolitik als MySQL. Sicherheitshinweise werden transparent mit der jeweiligen CVE-Identifikation bekanntgegeben. Durch die

[149] Vgl. **[MarLF]**
[150] Angelehnt an **[MarAP]**
[151] Vgl. ebd.
[152] Vgl. **[MarASV]**

zügige Veröffentlichung von Sicherheitsupdates wird die Bedrohung durch Softwarefehler verringert.[153] Die Aktualisierungen werden als Major- oder Minor Version bzw. als Patch Number veröffentlicht.[154] Genauso wie bei MySQL müssen die Sicherheitsupdates manuell eingespielt werden. Für Major- und Minor Versions (z.B. von Version 10.0 zu 10.1) muss das MariaDB-Datenbanksystem jedes Mal komplett deinstalliert und in neuster Version wieder installiert werden. Jede Produktversion wird über einen Zeitraum von 5 Jahre mit Updates unterstützt.[155]

5.2.7 Sicherheitsvorfälle

Die Anzahl von MariaDB-Sicherheitsvorfällen (siehe Tabelle 6) wurden durch eine Auswertung der NVD-Daten der letzten 5 Jahre (Auswertezeitraum 01. Januar 2011 bis 31. Dezember 2015) auf Basis des Stichworts „MariaDB“ generiert.

Jahr	Niedrig (0-3)	Mittel (4-6)	Hoch (7-10)	Gesamt
2011	0	1	0	1
2012	11	46	3	60
2013	7	36	0	43
2014	12	34	3	49
2015	13	25	2	40
Gesamt	43	142	8	193

Tabelle 6 - MariaDB Sicherheitsvorfälle

In dem ausgewerteten Zeitraum wurden insgesamt 193 Schwachstellen des MariaDB-Datenbanksystems registriert. Die meisten Schwachstellen sind in beiden Datenbanksystemen enthalten. Wie bei MySQL betrifft die häufigste auftretende Schwachstelle die Einschränkung der Vertraulichkeit, der Integrität sowie der Verfügbarkeit.

5.3 PostgreSQL

PostgreSQL ist ein objektrelationales Datenbanksystem, das sich hinter dem Bekanntheitsgrad von MySQL nicht verstecken muss. Das Datenbanksystem wurde in der Vergangenheit durch mehrere Preise ausgezeichnet.[156] PostgreSQL ist unter der PostgreSQL-Lizenz erhältlich, die ähnlich der Berkeley Software Distribution License (BSD-License) oder der Massachusetts Institute of Technology License (MIT-License) vertrieben wird. Die Lizenzen stammen aus dem Open-Source-Bereich und gestatten das Kopieren, Verändern sowie Verarbeiten unter dem Copyright-Vermerk der Original-Software.[157] Ein weiterer Ableger von PostgreSQL ist die kommerzielle Distribution „Postgres Plus Advanced Server“ des

[153] Vgl. **[MarSV]**
[154] Vgl. **[Ken15]** Kap. 1, S. 2
[155] Vgl. **[MarMP]**
[156] Vgl. **[PosAW]**
[157] Vgl. **[PosL]**

Anbieters EnterpriseDB. PostgreSQL entstand aus dem POSTGRES-Projekt, das an der University of California in Barkeley im Jahre 1986 unter der Leitung von Professor Michael Stonebraker entwickelt wurde. 1989 wurde die erste Version von POSTGRES veröffentlicht. Im Jahr 1993 wurde das POSTGRES-Projekt auf Grund der großen Anzahl an Support-Anfragen in der Version 4.2 eingestellt. Durch die beiden Studenten Andrew Yu und Jolly Chen wurde im Jahr 1994 das Projekt unter neuen Namen „Postgres95“ wiederbelebt und um einen SQL-Interpreter erweitert. 1996 kam es zu einer erneuten Änderung des Namens – aus Postgres95 wurde PostgreSQL Version 6.0.[158] Die Beliebtheit und das große Interesse am Open-Source-Datenbanksystem spiegelt sich in der permanenten Weiterentwicklung wieder. Aktuell liegt PostgreSQL in der Version 9.5.1 vor.

5.3.1 Konfiguration

PostgreSQL ist für die Plattformen BSD, Linux, Mac OS X, Solaris und Microsoft Windows erhältlich. Unter Microsoft Windows wird nach dem Herunterladen der Installationsdatei (postgresql-9.5.1-1-windows.exe) die Installationsroutine gestartet und die Konfiguration erfolgt über eine grafische Benutzeroberfläche.[159] Unter Linux können die PostgreSQL-Paketquellen direkt in der Konsole (Terminal) oder dem grafischen Paketmanager installiert werden. Die Installation verlangt unter Microsoft Windows die Eingabe eines neuen Passworts für den Superuser „postgres“, die Bestätigung oder Änderung der Portnummer für den Standarddienst (Port 5432) sowie die Eingabe der Länderregion (z.B. DE) zur Festlegung der länderspezifischen Zeichencodierung und des Nummernformates. Unter Linux wird der Superuser „postgres“ ohne Passwortschutz erzeugt. Der Netzwerkzugriff ist in der Standardkonfiguration auf das lokale Netzwerk (Localhost) beschränkt. Der Administrator muss nach der Installation die Konfiguration des Datenbanksystems in der Konfigurationsdatei postgresql.conf manuell mit Hilfe eines Texteditors anpassen und erweitern. Mit Hilfe des Open-Source-Tools pgAdmin kann die Konfiguration auch über eine grafische Administrationsoberfläche erfolgen.[160]

Die Verbindungsverschlüsselung ist wie bei MySQL nicht standardmäßig aktiviert, da zuerst die persönlichen SSL/TLS-Zertifikate erzeugt werden müssen. Weiter verfügt PostgreSQL über eine Testdatenbank „postgres“, die für Testzwecke standardmäßig aktiv ist.[161] In der Standardkonfiguration wird das Auditing aktiviert und protokolliert alle Aktivitäten am Datenbanksystems.

[158] Vgl. **[Pos15]** S. lxv- lxvi
[159] Vgl. **[PosD]**
[160] Vgl. **[Eis13]** Kap.2, S. 23
[161] Vgl. **[Pos15]** Kap. 17.2, S. 435

5.3.2 Authentifizierung

PostgreSQL bietet eine große Vielfalt an Authentifizierungsmethoden an. Die Identität von Clients wird anhand der Hostadresse, des Datenbanknamens und dem Benutzernamen überprüft.[162] Die host-basierte Authentifizierung (Host-Based Authentication) wird durch die Konfigurationsdatei pg_hba.conf gesteuert. Diese Konfigurationsdatei wird beim Start des Datenbanksystems in den Arbeitsspeicher geladen. Änderungen an der Konfigurationsdatei sind erst nach einem Neustart oder nach Eingabe des Befehls pg_ctl reload wirksam. Jede Zeile der HBA-Konfigurationsdatei beinhaltet Einträge in Form von: Verbindungstyp (TYPE), Datenbankname (DATABASE), Benutzername (USER), Client-Hostadresse (ADRESS)und Authentifizierungsmethode (METHOD).[163]

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# Database administrative login by Windows with MD5 protection
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5

# Database administrative login by Unix domain socket
local   all             postgres                                peer
```

Abbildung 6 - PostgreSQL pg_hba.conf

In der Standardkonfiguration unterstützt Microsoft Windows die Password Authentication Methode (md5) und unter Linux das Peer Authentication Verfahren (peer). Die beiden Authentifizierungsmethoden akzeptieren in der Standardkonfiguration nur lokale Netzwerkverbindungen. Das Passwort wird bei beiden Authentifizierungsmethoden als Hashwert in der Datenbank gespeichert. Das zu verwendende Authentifizierungsverfahren wird in der Konfigurationsdatei (Spalte METHOD) durch Angabe von vorgegebenen Kürzeln spezifiziert (siehe Abbildung 6). Neben der Standardauthentifizierung unterstützt das PostgreSQL-Datenbanksystem weitere Authentifizierungsverfahren.

- Password Authentication (siehe Kapitel 4.2.1)
- PAM Authentication (siehe Kapitel 4.2.2)
- Certificate Authentication (siehe Kapitel 4.2.3)
- LDAP Authentication (siehe Kapitel 4.2.4)
- RADIUS Authentication (siehe Kapitel 4.2.6)
- Trust Authentication
- GSSAPI Authentication

[162] Vgl. **[Pos15]** Kap. 19, S. 513
[163] Vgl. ebd. Kap. 19.1, S. 513-519

- SSPI Authentication
- Ident Authentication
- Peer Authentication

Die Trust Authentication Methode (trust) gewährt ohne Überprüfung von Benutzernamen (USER) und Passwort eine Verbindung. Die Zugriffsbeschränkungen werden aber weiterhin kontrolliert. Das Authentifizierungsverfahren sollte nur bei Einzelplatzsystemen (Single-User Mode) im lokalen Betrieb oder bei Mehrbenutzersysteme (Multi-User Mode) verwendet werden, wenn ausreichender Zugriffsschutz vom Betriebssystem (z.B. Unix-Domain Socket) gewährleistet werden kann.[164]

Das GSSAPI und das Security Support Provider Interface (SSPI) sind Programmierschnittstellen, die ohne Verwendung einer weiteren Authentifizierungsmethode (z.B. Kerberos) keine bzw. nur eine schwache Sicherheit bieten. Die GSSAPI Authentication (gss) wird ausschließlich bei Linux-Betriebssystemen und die SSPI Authentication (sspi) nur unter Windows-Betriebssystemen verwendet.[165]

Die Ident-based Authentication (ident) beruht auf einem Identifikationsprotokoll, das betriebssystemseitig einen Dienst (identd) bereitstellt, um Benutzer direkt am Betriebssystem zu authentifizieren. Beim Anmelden an das Datenbanksystem wird der DB-Benutzername mit Hilfe des Ident-Dienstes auf Gleichheit überprüft. Bei einer Übereinstimmung erfolgt dann die Authentifizierung am Datenbanksystem.[166]

Das Peer Authentication Verfahren (peer) nutzt zur Authentifizierung den Benutzernamen des Clients aus dem Kernel des Linux-Betriebssystems und vergleicht ihn mit den möglichen Datenbankbenutzern. Die Authentifizierung ist nur im lokalen Netzwerk möglich.[167]

5.3.3 Zugriffskontrolle

Für die Identitäts- und Autorisierungsverwaltung steht in PostgreSQL die rollenbasierte Zugriffskontrolle zur Verfügung. Neben der Zugriffssteuerung werden hierbei auch die Berechtigungen auf Objekte wie z.B. Schemata (beschreibt den Namensraum innerhalb einer Datenbank), Tabellen und Spalten gesteuert. Die Zugriffsrechte von Benutzern werden nach dem Konzept von Rollen verwaltet. Eine Rolle kann entweder direkt einem Benutzer oder einer Benutzergruppe zugewiesen werden. Die Rollen lassen sich auch ineinander verschachteln, so dass einem Benutzer die Berechtigungen auf eine andere Benutzerrolle gestattet

[164] Vgl. **[Pos15]** Kap. 19.3.1, S. 521
[165] Vgl. ebd. Kap. 19.3.3, S. 521-523
[166] Vgl. ebd. Kap. 19.3.5, S. 523-524
[167] Vgl. ebd. Kap. 19.3.6, S. 524

wird (Mitglied von). Hinzu können Rollen auch Datenobjekte beherbergen und den Zugriff von anderen Rollen auf diese Objekte regeln.[168] Die Abbildung 7 verdeutlicht beispielshaft die Vergabe von rollenbasierten Zugriffsrechten bei PostgreSQL.

```
postgres=# CREATE USER Student;
CREATE ROLE
postgres=# CREATE ROLE WInf;
CREATE ROLE
postgres=# GRANT WInf TO Student;
GRANT ROLE
postgres=# ALTER ROLE Student CREATEDB;
ALTER ROLE
postgres=# \du
                                Liste der Rollen
 Rollenname |                         Attribute                         | Mitglied von
------------+-----------------------------------------------------------+-------------
 postgres   | Superuser, Rolle erzeugen, DB erzeugen, Replikation | {}
 student    | DB erzeugen                                               | {winf}
 winf       | kann nicht einloggen                                      | {}
```

Abbildung 7 - PostgreSQL Zugriffskontrolle

Weiter können die Objektberechtigungen (GRANTS und REVOKE) auf Datenbankebene, Schemaebene, Tabellenebene und Spaltenebene weiter verfeinert werden. Dazu wird jede Ebene mit speziellen Privilegien (z.B. Select, Delete) belegt und allen Benutzer (Public), einer Benutzergruppe oder einer Rolle zugewiesen. Dies ermöglicht eine sehr granulierte Vergabe der Berechtigungen. Die Berechtigungen auf Objekte werden über die zentrale DAC-Zugriffskontrolle gesteuert.[169] Im Einzelnen können: Datenbanken separat abgesichert werden, Rechte auf Schemata vergeben werden, Tabellen abgesichert werden und auch einzelne Spalten geschützt werden.

Die Sicherheitsstrategie kann in PostgreSQL auf Statements (Objekte) und Rollen festgelegt werden. Mit Hilfe dieser auf Objekt- und Rollenbasis definierbaren Zugriffsrechten bietet PostgreSQL die Grundlage für die Umsetzung einer guten Sicherheitspolitik.[170]

Die Zugriffserweiterung Security Enhanced PostgreSQL (Sepgsql) ist eine Label Based Mandatory Access Control, speziell für Linux-Betriebssysteme, die das Security Enhanced Linux (SELinux) unterstützen. SELinux bezeichnet ein Zusatzmodul im Linux-Kernel, der die Basis für eine regelbasierte Zugangskontrolle bereitstellt. Durch die erweiterte Sicherheitsebene werden alle Datenobjekte (z.B. Tabellen, Spalten oder Tupele) mit der Systemzugriffsberechtigung des Betriebssystems überprüft. Dies findet durch

[168] Vgl. **[Pos15]** Kap. 20, S. 530
[169] Vgl. ebd. S. 1524-1530
[170] Vgl. **[PosRSP]**

Zuweisung von Security Labels auf Datenobjekte statt, die zusätzlich zur rollenbasierten Zugriffskontrolle agieren.[171] Durch die Informationsflusssteuerung zwischen Datenbanksystem und Betriebssystem lassen sich sensitive Daten besser vor unberechtigter Veränderung (Defacing), Zerstörung (Destruction) oder ungewollter Veröffentlichung (Leaking) schützen.[172]

5.3.4 Verschlüsselung

Das Datenbanksystem PostgreSQL stellt wie MySQL ausschließlich die Datenverschlüsselung auf Spaltenebene (Column-Level Encryption) bereit.[173] Das pgcrypto-Modul steuert die kryptographischen Funktionen des PostgreSQL-Datenbanksystems. Vor der Verwendung muss das Modul durch den Befehl CREATE EXTENSION pgcrypto aktiviert werden. Das pgcrypto-Modul verschlüsselt die Nutzdaten und Passwörter in der Datenbank.[174] Die unterstützten Verschlüsselungsmethoden des pgcrypto-Modul umfassen die Passwort-Hashfunktionen, die Pretty Good Privacy (PGP) Verschlüsselung und die Rohdaten (RAW) Verschlüsselung.

Den Passwort-Hashfunktionen stehen in PostgreSQL die Funktionen crypt(), die den gewünschten Hash-Algorithmus definiert und die Funktion gen_salt(), die die Verschlüsselungsstärke definiert, zur Verfügung. Neben den Standard-Hashfunktionen MD5, SHA-1 sowie SHA-2 (z.B. SHA-224, SHA-256, SHA-384, SHA-512) unterstützt PostgreSQL zusätzlich noch die Algorithmen Blowfish, DES und Extended DES für die Passwortverschlüsselung.[175]

Die PGP-Verschlüsselung beruht auf dem OpenPGP-Standard, der sich aus dem asymmetrischen Verschlüsselungsverfahren sowie dem Public-Key-Verschlüsselungsverfahren zusammensetzt. Das Public-Key-Verfahren stützt sich auf eindeutige Schlüsselpaare, die einen öffentlichen Schüssel (Public Key) und einen privaten geheimen Schlüssel enthalten. Bei der Verschlüsselung mit dem Public-Key-Verfahren wird ein neuer zufälliger Sitzungsschlüssel (Session Key) erzeugt, der durch den öffentlichen Schüssels signiert wird. Der PGP-Verschlüsselung stehen die Algorithmen Blowfish, AES (z.B. AES-128, AES-192 und AES-256), 3DES sowie CAST5 zur Verfügung.[176]

Die RAW-Verschlüsselung verwendet den lokalen Datenbankschlüssel des jeweiligen Datenbankbenutzers direkt als Schlüssel für die Chiffrierung der Rohdaten. Das RAW-Verfahren verfügt über keine Integritätsprüfung, ob

[171] Vgl. **[Pos15]** Kap. F.34, S. 2811
[172] Vgl. **[SEP]**
[173] Vgl. **[Pos15]** Kap. 17.8, S. 451-452
[174] Vgl. ebd. Kap. F.25, S. 2778-2789
[175] Vgl. ebd. Kap. F.25.2, S. 2778-2780
[176] Vgl. ebd. Kap. F.25.3, S. 2781-2786

verschlüsselte Daten modifiziert wurden. Unterstützt werden der Blowfish- und der AES-Algorithmus mit der internen Betriebsart CBC.[177]

5.3.5 Auditing

Das PostgreSQL-Datenbanksystem stellt mehrere Instrumente zur Überwachung und Protokollierung von Datenbankaktivitäten zur Verfügung. Standardmäßig werden alle Ereigniseinträge in einer zentralen Logdatei protokolliert. In der Konfigurationsdatei (postgresql.conf) können unter dem Konfigurationspunkt „Error Reporting and Logging“ die Ausgabe sprich „Wohin soll geloggt werden“ (Where to log), das Erscheinungsbild „Was soll geloggt werden“ (What to log) und das zeitbedingte „Wann soll geloggt werden“ (When to log) konfiguriert werden.[178]

Im Abschnitt „Wohin soll geloggt werden“ kann zwischen der Standardfehlerausgabe (Stderr), dem zentralen Ereignisprotokolldienst Syslog unter Unix-Betriebssystemen, dem Ereignisprotokoll unter Microsoft Windows (Eventlog) oder einer Dateiausgabe im Komma getrennte Dateiformat CSV (Csvlog) ausgewählt werden.[179] Weiter lässt sich noch das Verzeichnis der Logdatei, der Platzhalter im Dateinamen für Datum und Uhrzeit (standardmäßig postgresql-%Y-%m-%d_%H%M%S.log) sowie der Wechsel zwischen den Logdateien (Logfile Rotation) durch die Zeitbegrenzung einer Logdatei in Minuten oder der maximalen Dateigröße in Kilobytes festlegt werden. Wird die Zeitbegrenzung oder max. Dateigröße überschritten, wird eine neue Logdatei erzeugt.[180]

Mit Hilfe der Konfiguration „Wann soll geloggt werden“ lässt sich die Detailstufe (Message Security Levels), also der Informationsgehalt der zu protokollierten Ereignisse, steuern. Die Detailstufen sind in Debug5 bis Debug1, Info, Notice, Warning, Error, Log, Fatal und Panic unterteilt. Je niedriger die Einstufung der Detailstufe ist, desto höher ist der Informationsumfang, der in der Logdatei protokolliert wird. Weiter lässt sich kontrollieren, welchen Informationsgehalt jeder Benutzer bei einem Datenbankfehler sieht, welche minimale Detailstufe in der Logdatei protokolliert wird oder welches SQL-Statement einen Fehler verursacht hat bzw. nach welcher vorgegebenen Ausführungszeit eine SQL Anweisung in der Logdatei festgehalten wird. In der Standardkonfiguration werden Benachrichtigungen bis zur Detailstufe Notice an den Client gesendet, minimale Warnungen (Warnings) von möglichen Problemen protokolliert und Fehler (Errors), die zu Abbrüchen führen, dokumentiert. Die Standardeinstellung nach der Datenbankinstallation ist die Detailstufe Error; d.h., alle Ereignisse, die einen

[177] Vgl. **[Pos15]** ebd. Kap. F.25.4, S. 2786-2787
[178] Vgl. ebd. Kap. 18.8, S. 484-493
[179] Vgl. **[Eis13]** Kap. 2, S.50-55
[180] Vgl. **[Pos15]** Kap. 18.8.1, S.484-487

Fehler versuchen, wie auch alle Ereignisse der Detailstufe Fatal Errors und Panic werden in den Logdateien protokolliert. Die Zeitüberschreitung (Time Duration) einer Anweisung ist in der Standardkonfiguration deaktiviert.[181]

Weiterhin wird der Inhalt der Logdateien durch den Abschnitt „Was soll geloggt werden“ bestimmt. Die wichtigsten Ereignisse sind der Verbindungsaufbau und –abbau, die Protokollierung von Ausführungen von DDL- und DML-Kommandos sowie in der Entwicklungsphase und im Probebetrieb die Debug-Funktion. Weiter können externe Anwendungen, die eine Verbindung zur Datenbank benötigen, intern mit einem Namen belegt werden, um im Fehlerfall die Ursache leichter in der Logdatei zu lokalisieren.[182]

Zur Überwachung (Monitoring) der Datenbankaktivitäten und für die Performanceanalyse stellt PostgreSQL eine Sammlung von Statistiken zur Verfügung, die über die pg_stat_ und pg_statio_Systemsichten ausgegeben werden kann. Die Sammlung von Monitoring- und Performancedaten kann in der Konfigurationsdatei angepasst werden, um die Erhebung der erforderlichen Analysedaten für die Statistik zu gewährleisten. Standardmäßig werden alle ausgeführten Anweisungen jedes einzelnen Datenbankbenutzers mit der genauen Uhrzeit protokolliert. Dabei wird automatisch eine Statistik der Datenbankaktivitäten angelegt. Die Zeitmessung (Timing) der einzelnen ausgeführten Aktivitäten ist standardmäßig deaktiviert, da es bei manchen Betriebssystemen durch die ständige Abfrage zu Performanceeinbußen kommen kann. Weiter kann die Anzahl von Funktionsaufrufen sowie die Ausführungszeit (z.B. bei prozeduralen Sprachen oder SQL) aufgezeichnet werden. Für die Ausgabe der Statistik stehen PostgreSQL neben den internen Funktionen zahlreiche weitere Open-Source-Werkzeuge (z.B. Open PostgreSQL Monitoring) zur Verfügung.[183]

5.3.6 Sicherheitsupdate

Die PostgreSQL Global Development Group (PGDG) veröffentlicht Sicherheitsupdates anders als die Oracle Corporation zu der MySQL zählt in nicht vorgeschriebenen Abständen. Darüber hinaus werden die Sicherheitsprobleme auf der Produktseite direkt mit der entsprechenden CVE-Identifikation gelistet. Die Sicherheitsupdates für das PostgreSQL-Datenbanksystem werden in der Regel ohne Versionsaktualisierung (Minor Version Upgrade) veröffentlicht.[184] Die Versionsaktualisierungen ohne Produkterneuerungen (identisch mit Maintenance Releases) sind an der Inkrementierung der Versionsnummer an der dritten Stelle

[181] Vgl. **[Pos15]** Kap. 18.8.2, S.487-489
[182] Vgl. ebd. Kap. 18.8.3, S. 489-492
[183] Vgl. ebd. Kap. 18.9.1, S. 493-494
[184] Vgl. **[PosSI]**

(z.B. Version 9.2.3 zu 9.2.4) zu erkennen und enthalten ausschließlich Fehlerbehebungen (Bug Fixes). Softwareaktualisierungen (Major Release) sind an der Erhöhung der ersten bzw. zweiten Stelle der Versionsnummer erkennbar (z.B. Version 9.2 zu 9.3) und werden bei PostgreSQL jährlich veröffentlicht. Jede einzelne Produktversion wird über einen Zeitraum von 5 Jahren mit Sicherheitsupdates unterstützt.[185]

Sicherheitsupdates müssen in PostgreSQL-Datenbanksystem manuell eingespielt werden. Fehlerbehebungen, die als Minor Version veröffentlicht werden und ausschließlich Schwachstellen beheben, können direkt ohne großen Aufwand eingespielt werden. Für Softwareaktualisierungen (Major Release) die Änderungen an der Struktur der Systemtabellen beinhalten, steht die Funktion pg_upgrade zur Verfügung, die die Migration des Datenbanksystems unterstützt. Bei Änderungen am Datenformat bzw. am Speichersubsystem (Storage-Engine) stößt diese Funktion aber auch an ihre Grenzen.[186] Wie auch bei anderen Datenbanksystemen ist es notwendig vor einer Aktualisierung den kompletten Datenbestand zu sichern und später in die neu installierte Datenbankumgebung zu integrieren.

5.3.7 Sicherheitsvorfälle

Die Anzahl von PostgreSQL-Sicherheitsvorfällen (siehe Tabelle 7) wurden durch eine Auswertung der NVD-Daten der letzten 5 Jahre (Auswertezeitraum 01. Januar 2011 bis 31. Dezember 2015) auf Basis des Stichworts „PostgreSQL“ generiert.

Jahr	Niedrig (0-3)	Mittel (4-6)	Hoch (7-10)	Gesamt
2011	0	3	0	3
2012	0	9	1	10
2013	0	6	3	9
2014	1	12	2	15
2015	0	5	2	7
Gesamt	1	35	8	44

Tabelle 7 - PostgreSQL Sicherheitsvorfälle

In dem ausgewerteten Zeitraum wurden lediglich 44 Schwachstellen im PostgreSQL-Datenbanksystem bekannt. Wiederkehrende Schwachstellen sind z.B. SQL Injection, Buffer Overflows und Fehler im pgcrypto-Verschlüsselungsmodul.

[185] Vgl. **[PosVP]**
[186] Vgl. **[PosPG]**

5.4 Oracle Database 12c

Das Oracle Datenbanksystem zählt zu den Marktführern im Segment der kommerziellen relationalen Datenbanksysteme und ist speziell auf große Datenbestände und Anwendungen zugeschnitten. Die Zielgruppe des Oracle Datenbanksystems sind mittlere bis große Geschäftskunden. Das Datenbanksystem ist in unterschiedlichen Varianten erhältlich, die speziell auf die diversen Anwendungsfelder und Entwicklungsszenarien zugeschnitten sind. Die Produktpalette reicht von der freien Oracle Database Express Edition, über die Oracle Standard Edition 2 bis hin zur Oracle Database Enterprise Edition.[187] Die im Funktionsumfang eingeschränkte Express Edition ist kostenfrei nutzbar, liegt aber aktuell nur in der veralteten Version 11g vor. Die anderen Editionen unterscheiden sich prinzipiell im Funktionsumfang und den damit verbundenen Lizenzkosten. Je Prozessor (CPU) belaufen sich die Lizenzkosten auf 17.500 US Dollar für die Standard Edition 2 bis hin zu 47.500 US Dollar für die Enterprise Edition pro Jahr (Stand: 1. Dezember 2015).[188]

Der Ursprung des Oracle Datenbanksystem reicht bis in die späten 70er Jahre zurück. Damals überprüfte Lawrence „Larry“ J. Ellsion routinemäßig die Forschungsarbeiten der International Business Machines Corporation (IBM) und entdeckte die wissenschaftliche Arbeit „System R: Relational Approach to Database Management“, die einen funktionierenden Prototyp eines relationalen Datenbankmanagementsystems beschrieb. Ellison zeigte die Ausarbeitung seinen Kollegen Bob Miner und Ed Oates, die gemeinsam bei Ampex arbeiteten. Das Trio erkannte das enorme Geschäftspotenzial hinter relationalen Datenbanken und gründeten im Jahr 1977 gemeinsam das Unternehmen Software Development Laboratories (SDL). Im Jahr 1978 entwickelte das Trio das Datenbanksystem Oracle Version 1, das in Assemblersprache programmiert war und nie veröffentlicht wurde. Im Jahr 1979 erschien die Version 2, das erste kommerzielle relationale Datenbanksystem dieser Zeit. Im selben Jahr wurde der Unternehmensname in Relational Software Inc. (RSI) umbenannt. 1982, drei Jahre später, fand die erste öffentliche Konferenz des Oracle Datenbanksystem in San Francisco statt. Gleichzeitig wurde das Unternehmen auch in Oracle Systems umgetauft. Die Version 3, die 1983 erschien, war in der Programmiersprache C geschrieben und das erste relationale Datenbankmanagementsystem das für Großrechner, Minicomputer und Personal Computers (PC) zugeschnitten war. Im Jahr 1985 erschien die Oracle Version 5, das erste relationale Datenbanksystem, das in einer Client-Server-Umgebung lauffähig war. Das Datenbanksystem wurde 1989 um die Echtzeit-Transaktionsverarbeitung (Online-Transaction-Processing) erweitert.

[187] Vgl. **[OraSD]**
[188] Vgl. **[OraPL]**

Oracle war das erste Softwareunternehmen, das im Jahr 1993 Geschäftsprozesse für Client-Server-Anwendungen zentral in einem Rechenzentrum bereitstellte (Software-as-a-Service). Weiter stellte Oracle im Jahr 1995 als erstes Unternehmen eine umfassende Internetstrategie vor. Oracle gab im Jahr 1999 bekannt, dass alle Oracle Produkte (damals Oracle 8i Database) offene Standards und Technologien wie z.B. Linux oder die Extensible Markup Language (XML) unterstützen. Die Oracle 9i Database wurde 2001 um das Oracle Real Application Cluster erweitert, das den Kunden die Möglichkeit bot, die Datenbank auszulagern, um die Erreichbarkeit, die Skalierbarkeit und die Leistungsfähigkeit mittels verteilten kostengünstigen Servern (Low-Cost-Server) zu steigern (Clustering). 2003 debütierte die Oracle Database 10g, die in der Enterprise Version das Grid-Computing unterstützte und dadurch automatisch die Auslastung bedarfsgerecht verteilte. Im Jahr 2007 wurde die Oracle Database 11g vorgestellt, die bis heute noch große Verwendung findet. 2013 erschien der Nachfolger Oracle Database 12c, das in erster Linie das Cloud-Computing unterstützte und Database-as-a-Service ermöglichte. Im Jahr 2014 folgte die In-Memory-Option, um die Abfragezeit (Query Time) zu verbessern. Die Version 12c des Oracle Datenbanksystems zählt heute noch zum aktuellsten Entwicklungsstand.[189]

5.4.1 Konfiguration

Die Oracle Database 12c (12.1.0.2.0) ist direkt über die Oracle Produktseite erhältlich und unterstützt in der Standard- und Enterprise Edition die Betriebssysteme Microsoft Windows, Linux, Oracle Solaris (SPARC und x86 Systeme), HP-UX Itanium, AIX und zLinux64. Die lizenzfreie Oracle Database 11g Express Edition ist ausschließlich für die Betriebssysteme Linux und Microsoft Windows erhältlich.[190] Die Installationsdateien sind in zwei komprimierte Archivdateien aufgeteilt, die nach dem Download im ZIP-Dateiformat vorliegen. Vor der Installation müssen die Dateien erst entpackt und zusammengeführt werden. Der Oracle Universal Installer (OUI) ist ein Java-Programm, das plattformunabhängig die Installationsroutine einheitlich auf allen Betriebssystemen durchführt. Wahlweise stehen die interaktive Installationsroutine und die automatisierte Installationsmethode mittels einer Hilfsdatei zur Verfügung. Die automatisierte Installation eignet sich für die Bereitstellung von mehreren Datenbanksystemen mit derselben Grundeinstellung.[191]

Die interaktive Installationsroutine führt durch die gesamten Installationsschritte der Datenbankinstallation. Im ersten Schritt werden die Sicherheitsaktualisierungen konfiguriert. Hierzu wird die eigene E-Mail-Adresse und das Kennwert für den

[189] Vgl. **[Ora07]**
[190] Vgl. **[OraSD]**
[191] Vgl. **[OraDI]**

Oracle Support benötigt, andernfalls ist das Datenbanksystem ungepatcht und erhält auch während des Betriebs keine Softwareaktualisierungen. Im nächsten Installationsschritt werden die Installationsoptionen gewählt. Hier steht zur Auswahl eine „Datenbank zu erstellen und zu konfigurieren“, „Nur Datenbanksoftware installieren“ sowie ein „Upgrade einer bestehenden Datenbank“ durchzuführen. Im dritten Schritt, nach der Auswahl „Datenbank zu erstellen und zu konfigurieren“, wird definiert, in welcher Systemklasse das Datenbanksystem zur Anwendung kommt. Dafür stehen die Optionen „Desktopklasse“ mit minimaler Konfiguration und einer Startdatenbank zur Verfügung sowie die „Serverklasse“ für den professionellen Betrieb mit erweiterten Konfigurationsumfang (z.B. Oracle Real Application Cluster, Automatic Storage Management, Backup und Recovery). Im Abschnitt vier wird die Datenbankinstallationsart gewählt. Zur Auswahl stehen: „Datenbankinstallation mit nur einer Instanz“, „Oracle Real Application Clusters-Datenbankinstallation“ und „Installation von Oracle One Node-Datenbank“.

Der fünfte Schritt der Installation bietet die Möglichkeit zwischen der „Standardkonfiguration“ und der „Erweiterten Installation“ zu wählen. Die erweiterte Installation erzeugt z.B. unterschiedliche Kennwörter für die System-Accounts, automatisierte Backups oder alternative Speicheroptionen. Nach der Auswahl des Installationstyps „Erweiterte Installation“ erfolgt die Definition der Produktsprache. Standardmäßig sind die Sprachpakete Deutsch und Englisch vordefiniert. Der siebte Schritt verlangt die Auswahl der Datenbank-Edition. Da explizit die Enterprise-Edition heruntergeladen wurde, sind alle anderen Edition des Datenbanksystems nicht auswählbar und ausgegraut. Der achte Schritt ermöglicht die Wahl eines Benutzerkontos für den Oracle Home-Benutzer. Hier ist es wie bei allen anderen Datenbanksystemen essenziell dem Datenbankdienst (Service) keine zu hohe Berechtigung (z.B. Administrator) zukommen zu lassen. Der nächste Schritt der Konfiguration definiert das Installationsverzeichnis des Oracle Base-Verzeichnisses (Konfigurationsdateien) sowie das Softwareverzeichnis für die Oracle Database. Der zehnte Schritt legt die Verwendungsart fest, unterschieden wird die „Allgemeine Verwendung mit Transaktionsverarbeitung“ und dem „Data Warehousing“.

Im Konfigurationspunkt 11 wird der Datenbank-Identifier anhand des globalen Datenbanknamens und dem System Identifier bestimmt. Standardmäßig ist in beiden Fällen der Name „orcl“ vordefiniert. Die Instanz wird für die eindeutige Kennung benötigt, wenn mehrere Instanzen auf dem Datenbanksystem laufen. Der Abschnitt 12 bestimmt die maximale verwendbare Größe des Datenbankspeichers (RAM), den speziellen Zeichensatz und die Option, ein Beispielschema in der Startdatenbank zu integrieren. In Schritt 13 lässt sich ein optionaler Speicherort der Datenbankdateien wählen, um die Effektivität durch unterschiedliche Datenträger

zu erhöhen. Die Oracle Database 12c wird normalerweise bei Einzellösungen von Oracle Enterprise Manager Database Express verwaltet; alternativ kann der Enterprise Manager Cloud Control gewählt werden. Schritt 15 bietet die Möglichkeit optional die Wiederherstellung (Recovery) zu aktiveren, um einen Wiederherstellungspunkt zu setzen. Der letzte Schritt der Konfiguration verlangt die Eingabe der Passwörter für die Systemkonten (z.B. SYS, SYSTEM, DBSNMP, PDADMIN) zur Datenbankverwaltung. Die Installationsroutine warnt vor zu schwach bzw. zu einfallslosen Passwörtern.

In Schritt 17 erfolgt die Überprüfung der Konfiguration und die anschließende Zusammenfassung. Zu diesem Zeitpunkt können noch Änderungen an den zuvor festgelegten Einstellungen vorgenommen werden. Darauf erfolgt die Installation des Oracle Datenbanksystems. Nach der erfolgreichen Installation stehen der Oracle Datenbanksystem weitere grafische Anwendungen (z.B. Administrationsassistent, Datenbank-Konfigurationsassistent oder der Wallet Manager) für die erweiterte Konfiguration und Absicherung des Datenbanksystems zur Verfügung. Der Oracle Enterprise Manager, der über den integrierten WebLogic Server erreichbar ist, stellt eine erweiterte Webschnittstelle zur Überwachung, Kontrolle und Performanceanalyse bereit.

5.4.2 Authentifizierung

Standardmäßig ist in allen Oracle 12c Datenbank Editionen die integrierte Benutzername-Passwort-Authentifizierung aktiv. Das Passwort wird während der Übertragung durch das Netzwerk mittels der automatischen und transparenten Passwortverschlüsselung, die auf dem AES-Algorithmus beruht, geschützt. Der Benutzername und das Passwort sind in der Systemtabelle dba_users der Datenbank gespeichert. Das Oracle Datenbanksystem unterstützt die Passwortgüte durch die Funktionen ora12c_verify_function und ora12c_strong_verify_function. Diese Funktionen überprüfen neue und geänderte Passwörter auf ausreichendenden Schutz und Komplexität hinsichtlich der Passwortqualität. Weiter ist die Benutzername-Passwort-Authentifizierung durch einen Mehrfach-Eingabe-Schutz abgesichert; nach jeder Falscheingabe erhöht sich die Wartezeit bis zur nächsten Eingabemöglichkeit. Seit dem Oracle 12c Release wird bei der Passworteingabe auch die Groß- und Kleinschreibung während der Authentifizierung überprüft.[192]

Geschützt wird das Passwort serverseitig in der Systemtabelle dba_users durch die kryptographische Hashfunktion SHA-2 mit der Schlüssellänge SHA-512 sowie der Password-Based Key Derivation Function 2 (PBKDF2). Das Passwort wird zuerst durch die kryptographische Hashfunktion verschlüsselt. Danach wird über

[192] Vgl. **[OraCA]**

mehrere Runden die Schlüsselableitungsfunktion PBKDF2 angewandt. Die Verkettung des Passwortes durch das Ableitungsverfahren erschwert das spätere „Erraten“ des ursprünglichen Passworts durch Brute-Force-Angriffe.[193]

Zu den starken Authentifizierungen (Strong Authentication) des Oracle 12c Datenbanksystems gehören die Authentifizierungsverfahren Kerberos, RADIUS und die Secure Sockets Layer Authentication.[194] Die externen Authentifizierungsmethoden der Netzwerksicherheit sind optional und können über das grafische Tool Oracle Net Manager oder manuell in der Konfigurationsdatei sqlnet.ora unter SQLNET.AUTHENTICATION_SERVICES = (KERBEROS5, RADIUS) konfiguriert werden.[195] Neben dem SSL-Algorithmus stehen wahlweise auch andere Algorithmen der Netzwerksicherheit zur Verfügung. Diese können durch den Net Manager unter Integrität (z.B. SHA, MD5) und Verschlüsselung (z.B. AES, 3DES, RC4) für den Client und Server ausgewählt werden.

Die Secure Sockets Layer Authentication (siehe Kapitel 4.2.7) baut im Oracle Datenbanksystem auf die PKI Authentifizierungsmethode mit öffentlichen Schlüsseln auf. Optional bietet die Oracle Advanced Security auf Grundlage der Cipher Suite weitere kombinierbare Verschlüsselungen für die Authentifizierung (z.B. RSA), die Verschlüsselung (z.B. RC4 128) und die Datenintegrität (z.B. SHA-1) an. Die einzelnen Methoden lassen sich entweder über das grafische Tool Oracle Net Manager oder direkt in der Konfigurationsdatei beliebig zusammenstellen.[196]

5.4.3 Zugriffskontrolle

Das Oracle 12c Datenbanksystem verfügt über eine Vielzahl von Zugriffskontrolloptionen. Der direkte Datenbankzugriff wird durch die rollenbasierte Zugriffskontrolle (RBAC) mittels Privilegien (Privileges) und Rollen gesteuert. Das RBAC-Modell gewährt einem Benutzer auf Grundlage der ihm zugewiesenen Rollen Zugriff auf ein gewünschtes Objekt. Die Berechtigungen sind in Systemprivilegien (System Privileges) und Objektprivilegien (Object Privileges) unterteilt. Die Berechtigungen auf Objekte werden durch die benutzerbestimmbare Zugriffskontrolle (DAC) gesteuert. Das DAC-Modell ermöglicht Besitzern eines Objekts die Berechtigungen (GRANTS und REVOKE) zu anderen Subjekten (Benutzer und Rollen) zu regeln.[197] Das Systemprivileg (Inherit Privileges) ermöglicht die Zuweisung von Rollen an PL/SQL-Objekte. Aufgerufene PL/SQL-Prozeduren laufen immer standardmäßig mit den Rechten des ausführenden

[193] Vgl. **[OraCA]**
[194] Vgl. **[OraSA]**
[195] Vgl. **[OraAM]**
[196] Vgl. **[OraSSL]**
[197] Vgl. **[Gae15]** Kap. 4

Benutzers ab. Wenn ein privilegierter Benutzer (z.B. DBA) eine PL/SQL-Prozedur mit dem Anhang AUTHID CURRENT_USER ausführt, kann es möglicherweise zu einer ungewollten Rechteausweitung kommen. Der Rechtekontext der Systemprivilegien wird durch die Code-Based Access Control (CBAC) geregelt. Dabei wird festgelegt, ob der PL/SQL-Code den Rechtekontext übernehmen bzw. zur Ausführung bringen darf.[198]

Die Oracle Virtual Private Database (VPD) auch geläufig als feingranulare Zugriffskontrolle (Fine Grained Access Control) stellt die Zugriffskontrolle auf Spalten- (Column) und Zeilenebene (Row Level) zur Verfügung. Dazu wird eine dynamische WHERE-Klausel an die SQL-Anfrage angehängt. Normale Objektprivilegien, die das Lesen, Schreiben, Einfügen, Ändern oder Löschen von Dateien ermöglichen, gelten immer für den kompletten Zeilenbereich einer Tabelle. Das VPD erweitert die Sicherheitsstrategie (Security Police) für den Zugriff auf einzelne Zeilen einer Tabelle, um den erweiterten Schutz sensitiver Daten zu ermöglichen. Diese Sicherheitsfunktion ist nur in der Oracle Enterprise Version nutzbar.[199]

Die Oracle Label Security (OLS) bietet eine integrierte Sicherheitspolitik, die wie VPD den Schutz auf Zeilenebene (Row-Level Security) ermöglicht. Anders als bei der VPD, wo PL/SQL-Code für die Sicherheitsfunktionen erstellt werden muss, stellt OLS eine Komplettlösung (Out-of-the-Box-Lösung) dar und erfordert keine zusätzliche Programmierung. OLS ist in der Hierarchie oberhalb von VPD angesiedelt (siehe dazu Abbildung 8). Mit OLS können Sicherheitsrichtlinien auf Zeilenebene durch einen speziellen Namen (Label) beschriftet werden. Durch die Beschriftung der Daten in unterschiedliche Empfindlichkeitsstufen (Sensitivity Levels) z.B. streng geheim (SG), geheim (G), vertraulich (V), öffentlich (Ö) und nicht klassifiziert (NK) können die sensitiven Daten in sogenannte Risikoklassen (Risk Levels) unterteilt werden. Diese Schutzart beruht auf der regelbasierten Zugriffskontrolle (MAC) und wird durch die Multi-Level Security nach der Vertraulichkeit unterteilt.[200]

[198] Vgl. **[OraIR]**
[199] Vgl. **[OraVPD]**
[200] Vgl. **[OraOLS]**

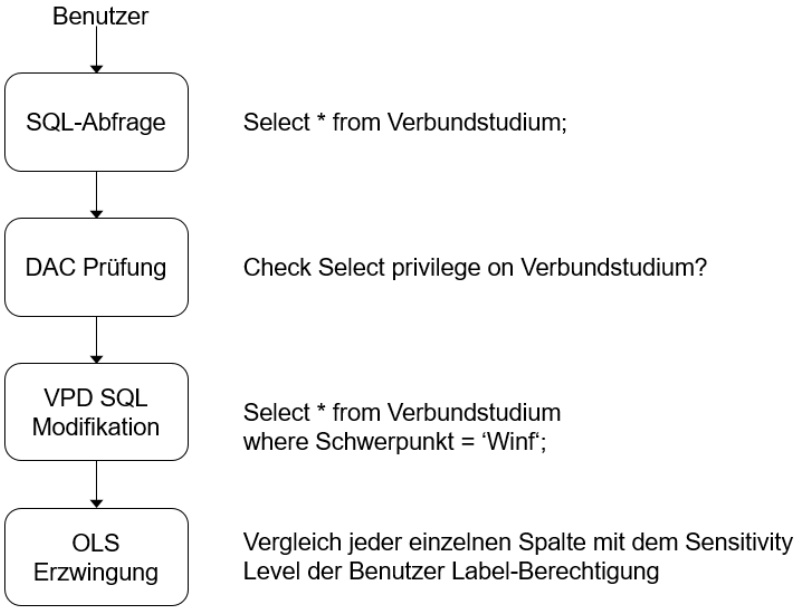

Abbildung 8 - Oracle Label Security Architektur[201]

Das Oracle Database Vault (DV) ist eine privilegierte Benutzerkontrolle und Analysewerkzeug, das in der Oracle 12c Enterprise Version standardmäßig vorkonfiguriert ist. Der Schutz beruht auf der regelbasierten Zugriffskontrolle (MAC) und war in den Vorgängerversionen (z.B. Oracle 11.1) nur gegen Aufpreis erhältlich. Durch den Datenbanktresor (Database Vault) lassen sich privilegierte Benutzerkonten wie z.B. die Datenbankadministratoren oder vordefinierte Systemkonten, die Zugang zu sämtlichen sensiblen Anwendungsdaten besitzen, einschränken und kontrollieren. Mit dem DV wird für die alltägliche Arbeit aufbauend auf der „Principle of least privilege“ Strategie eine stark eingeschränkte Anwendungsumgebung (Realm) bereitgestellt. Durch die Mandatory Realms können alle Daten (Tabellen, Ansichten oder Objekte) vor privilegierten Benutzern oder den Objekteigentümer geschützt werden, außer der Zugriffschutz wurde speziell für die jeweiligen Anwender gesetzt. DV ermöglicht auch die Kontrolle der Datenbankkonfiguration durch Überwachung der Befehle wie ALTER SYSTEM, ALTER USER, CREATE USER und DROP USER. Abweichungen an der Standardkonfiguration können schnell zu einem unstabilen und unsicheren Datenbanksystem führen, falls Änderungen an der Datenbankstruktur wie z.B. an Anwendungstabellen, den Rollen oder den privilegierten Benutzerkonten vorgenommen wurden. Generell lassen sich auch SQL- und System-Befehle überwachen. Zur Umsetzung einer konsequenten Rollentrennung (Separation of Duty) kann dem DBA die Benutzerverwaltung entzogen werden; die Verwaltung erfolgt dann durch den „Database Account Manager“ in einer separaten Rolle. In einem Realm können die Datenbankrollen durch Entzug der GRANT und REVOKE

[201] Angelehnt an **[OraLS]**

Befehle vor unautorisierten privilegierten Benutzern geschützt werden. Durch die Laufzeitanalyse (Run-time Privileg Analysis) werden unnötige Rollen und Privilegien aufgedeckt und angezeigt. Dadurch kann die minimale Privileg-Analyse für neu entwickelte oder bestehende Anwendungen sichergestellt werden. Die Berichtsfunktion des DV ermöglicht es, geblockte SQL-Befehle und Verletzungen von Sicherheitsrichtlinien durch den Realm Audit Report anzuzeigen.[202]

Die Oracle Real Application Security (RAS) wurde entwickelt, um ein umfassendes mit der Anwendung verbundenes Sicherheitsmodell auf Datenbankebene zu schaffen. RAS ist ein feinstufiges Berechtigungsmodell, das eine Anwendungssicherheitsumgebung durch Ende-zu-Ende-Sicherheit (End-to-end Security) schafft. Dazu werden Sicherheitsrichtlinien der Benutzerverwaltung, Rollenverwaltung, Datenbanksitzungsverwaltung im Anwendungskontext mit der feinkörnigen Datensicherheit (Privilegien und Berechtigungen) und dem Auditing für Anwendungen bereitgestellt.[203] Die Berechtigungen der RAS werden durch die Zugriffssteuerungsliste (Access Control List) und den Zugriffssteuerungseinträgen (Access Control Entries) gesteuert.[204]

5.4.4 Verschlüsselung

Für das Oracle Datenbanksystem stehen verschiedene Sicherheitsfunktionen zur Verschlüsselung von ruhenden und in der Ausgabe befindlichen Daten bereit.

Die transparente Datenverschlüsselung (TDE) ermöglicht den Schutz sensitiver Daten auf Spaltenebene (Column-Level Encryption) und Tabellenebene (Tablespace Encryption). Der Verschlüsselungsalgorithmus beruht auf AES mit der Schlüssellänge 128- oder 256-Bit sowie dem 3DES-Algorithmus mit der festen Schlüssellänge 168-Bit. Optional kann der Hash-Algorithmus SHA-1 mit einer Schlüssellänge von 160-Bit verwendet werden. Die Schlüsselverwaltung (Key Management) beruht auf einer Zweiwege-Schlüsselverwaltungsarchitektur mit einem Datenschlüssel und einem Master-Schlüssel. Der Datenschlüssel wird durch den Master-Schlüssel verschlüsselt und in der Datenbank automatisch verwaltet. Der Master-Schlüssel wird außerhalb der Datenbank im Oracle Wallet aufbewahrt. Die Schlüssel im Oracle Wallet sind durch den Public Key Cryptography Standards 12 (PKCS12) in einer Datei geschützt. Die Zweiwege-Schlüsselverwaltungsarchitektur ermöglicht das Austauschen des Masterschlüssels ohne die sensitiven Daten in der Datenbank erneut verschlüsseln zu müssen.[205]

[202] Vgl. **[OraDV]**
[203] Vgl. **[Gae15]** Kap. 6
[204] Vgl. **[OraRAS]**
[205] Vgl. **[OraAS]**

Zur Absicherung von ruhenden Daten, kann auch das Paket DBMS_CRYPTO genutzt werden. Nach entsprechender Konfiguration werden Inhalte auf Spaltenebene verschlüsselt. Der Benutzer muss dazu über die Ausführungsberechtigungen GRANTE EXECUTE verfügen. Das Paket stellt die kryptographischen Verfahren DES, 3DES, 3DES_2KEY (128-Bit Schlüssel), AES und RC4 sowie die Hashfunktionen MD4 (Vorgänger von MD5), MD5, SHA-1 und SHA-2 (SHA-256, SHA-384, SHA-512) bereit.[206]

Die Oracle Data Redaction ermöglicht es, sensible Daten für die Ausgabe durch Maskierung unkenntlich zu machen, teilweise bzw. gänzlich zu entfernen oder durch einen Standardwert zu ersetzen. Die ursprünglichen Daten bleiben dabei in der Datenbank unverändert erhalten. Die Aufforderung für das Unkenntlich-Machen der Daten wird dazu in der jeweiligen Tabelle hinterlegt und gilt ausschließlich für Datenabrufe von unautorisierten Benutzern, die für eine Druck- oder Bildschirmausgabe die Daten benötigt. Für unterschiedliche Benutzer oder Gruppen kann die Ausgabe unterschiedlich dargestellt werden. Z.B. ähnlich wie bei einem Kassenzettel, wo nur die letzten vier Ziffern der Kreditkartennummer (Primary Account Number) dargestellt werden.[207]

Die Oracle Data Redaction stellt vier Möglichkeiten zum Unkenntlich-Machen der Ausgabe bereit (siehe Tabelle 8).

- Full: Der komplette Wert wird durch eine Konstante ersetzt
- Partial: Ein Ausschnitt des Werts wird unlesbar gemacht
- RegExp: Der Ausdruck wird durch einen Platzhalter (Wildcard) ersetzt
- Random: Der gesamte Wert wird durch einen willkürlichen Wert ersetzt

	Stored Data	Redacted Display
Full	15.07.2016	01.09.2009
Partial	011-070-545	XXX-XXX-545
RegExp	Markus.Berg@smail.th-koeln.de	[hidden]@smail.th-kolen.de
Random	207604898495088	200000000000045

Tabelle 8 - Oracle Data Redaction[208]

5.4.5 Auditing

In den Versionen vor Oracle 12c war eine separate Protokollkette (Audit Trail) für jede individuelle Überwachungsfunktion vorhanden. Mit der Version 12c wurde mit Oracle Unified Auditing ein neuartiges Instrument zur Protokollierung und Überwachung von Datenbankaktivitäten bereitgestellt. Das Unified Auditing bündelt

[206] Vgl. **[OraC]**
[207] Vgl. **[Fab13]**
[208] Angelehnt an **[OraAS]**

alle Audit-Funktionen des Datenbanksystems und stellt die gesammelten Informationen zentral in einem Depot bereit.[209] Die Überwachungsrichtlinien (Audit Policies) des Unified Auditing sind Container, die verwendet werden können, um Aktionen, Privilegien oder Objekte zu auditieren. Diese Überwachungsrichtlinien sind frei konfigurierbar und können Rollen, Bedingungen oder Ausnahmen enthalten. Das Oracle Datenbanksystem enthält sieben vereinheitlichte Überwachungsrichtlinien (Unified Audit Policies):[210]

- ORA_LOGON_FAILURES Aufzeichnung fehlgeschlagener Anmeldeversuche (inaktiv)
- ORA_SECURECONFIG Überwachung der Audit Konfiguration (aktiv)
- ORA_DATABASE_PARAMETER Überwachung von Datenbank Parameter/SPFile (inaktiv)
- ORA_ACCOUNT_MGMT Überwachung von Benutzer/Rollen sowie die Vergabe von Rechten (inaktiv)
- ORA_CIS_RECOMMENDATIONS Überwachungsfunktion des Center for Internet Security (CIS) (inaktiv)
- ORA_RAS_POLICY_MGMT Überwachung der Ereignisse der RAS (inaktiv)
- ORA_DV_AUDPOL Überwachung der Aktionen der DV (inaktiv)[211]

Weitere Auditinformationen kommen unter anderem aus den Anwendungen RAS, DV, OLS, Oracle Data Mining und Oracle Data Pump sowie aus den Audit Records und den Fine-grained Audit Records. Die Auditinformationen werden aus den unterschiedlichen Quellen in einem einheitlichen Unified Auto Trial zusammengeführt. Die gesammelten Auditinformationen werden durch einen kryptischen Namen in der Tabelle AUDSYS.CLI_SWP*** im Schema AUDSYS gespeichert. Unified Auditing ist standardmäßig in der Oracle Enterprise Edition enthalten und installiert. Bei Neuinstallationen sind die herkömmlichen Auditing-Methoden (Mandatory Auditing, Standard Auditing und Fine Grained Auditing) und das neuartige Unified Auditing zusammen (Mixed Mode) aktiv (siehe Abbildung 9).[212]

[209] Vgl. **[Mil14]** S.4
[210] Vgl. **[OraAP]**
[211] Vgl. ebd.
[212] Vgl. **[OraIA]**

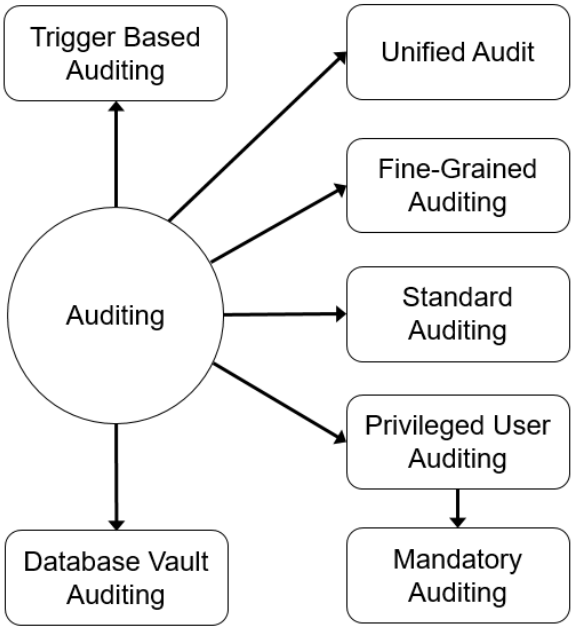

Abbildung 9 - Oracle Auditing[213]

In jedem Oracle Datenbanksystem ist das Mandatory Auditing aktiv; es lässt sich auch nicht deaktivieren. Neu ist ab der Version 12c, dass privilegierte Benutzerkonten wie z.B. SYSDBA auch der Überwachung unterliegen. Die überwachten Aktivitäten des Mandatory Auditing werden in der Tabelle SYS.UNIFIED_AUDIT_TRAIL gespeichert. Standardmäßig werden folgende prüfungsbezogene Aktivitäten überwacht: [214]

- CREATE/ALTER/DROP AUDIT POLICY (Überwachungsvorschrift)
- AUDIT/NOAUDIT
- EXECUTE der DBMS_FGA PL/SQL und DBMS_AUDIT_MGMT PL/SQL
- Konfigurationsänderungen, die an der Oracle DV durchgeführt werden
- Ausführen von Aktionen an der AUDSYS Tabelle
- TOP Level-Anweisungen von Administratoren (z.B. SYS, SYSDBA)[215]

Das Standard Auditing wird durch Setzen des Parameters Auto Trial aktiviert und durch die Befehle AUDIT/NOAUDIT konfiguriert. Um das Standard Auditing wirksam zu machen, muss die Datenbank neu gestartet werden. Dadurch lassen sich Statements (audit select table), Privilegien (audit select any table) und Objekte (audit select on BERG.WINF) überwachen. Weiter kann festgelegt werden, ob nur erfolgreiche oder nur fehlgeschlagene Aktionen dokumentiert werden. Die Art und Weise der Protokollierung lässt sich wahlweise für die Sitzung (by Session) oder für den Zugang (by Access) auswählen. Die Art der Protokollierung nimmt Einfluss

[213] Angelehnt an **[Oeh15]** S. 44
[214] Vgl. **[OraAT]**
[215] Vgl. ebd.

auf den Informationsgehalt. Oracle empfiehlt ausschließlich die Überwachung und Protokollierung mit der by Access Methode. Die Logdateien enthalten dadurch mehr Informationen von den durchgeführten Aktionen - wie z.B. den Ausführungsstatus (Return Code), das Datum und die Uhrzeit der Ausführung, das verwendete Privileg, die zugegriffenen Objekte, den SQL-Befehl selbst und die genutzten Werte.[216]

Das fein granulierte Auditing (Fine Grained Auditing) ermöglicht die Überprüfung bis auf die Spaltenebene (Column-Level). Um den Datenzugriff auf Spaltenelemente zu überwachen, müssen entsprechende Richtlinien definiert werden. Zur Erstellung einer fein granulierten Überwachungsrichtlinie (Audit Policy) wird die DBMS_FGA.ADD_POLICY Prozedur angewandt. Durch die Erweiterung kann festgelegt werden, welches Objekt unter welchen Bedingungen auditiert wird (siehe Abbildung 10). Des Weiteren können auch Bedingungen erstellt werden, die zuerst erfüllt sein müssen (z.B. Wertebereich), um einen Audit-Eintrag zu erzeugen. Aufrufe von DML-Befehlen wie INSERT, UPDATE und DELETE sowie auch SELECT können entsprechende Audit-Einträge generieren.[217]

```
BEGIN
  DBMS_FGA.ADD_POLICY (
    object_schema=>'WINF',
    object_name=>'STUDENT',
    policy_name=>'COMMUNITY_POLICY,
    audit_column=>'SEMESTER',
    audit_condition=>'SEMESTER < 9',
    statement_types=>'UPDATE',
    handler_schema=>'SYS',
    handler_module=>'FGA_USER_INFO',
    enable=>TRUE);
END;
/
```

Abbildung 10 - Oracle Fine Grained Auditing[218]

Das Auditing administrativer Konten (Privileged User Auditing) war in den Versionen vor 12c standardmäßig nicht aktiv und stellte ein großes Sicherheitsrisiko dar. Seit der Oracle Version 12c werden Aktionen administrativer Konten, zu denen auch SYSDBA oder SYSOPER zählen, defaultmäßig durch das Mandatory Auditing protokolliert. Durch das Setzen des Parameters

[216] Vgl. **[Dea15]**
[217] Vgl. ebd.
[218] Angelehnt an **[OraFGA]**

AUDIT_SYS_OPERATIONS = TRUE werden alle Aktionen des Benutzers SYS protokolliert und im Standard Audit Trail gespeichert.[219]

5.4.6 Sicherheitsupdate

Das Datenbanksystem liegt aktuell in der Version 12.1.0.2.0 vor. Die erste Zahl von links kennzeichnet die Hauptversionsnummer (Major Release Number) und wird bei einem neuen Versionsstand inkrementiert. Das „c“ in der verkürzten Schreibweise 12c steht für „cloud“ und besagt, dass das Datenbanksystem Cloud-Computing unterstützt. Der Schwerpunkt der Vorgängerversion 11g lag hingegen auf dem Grid Computing, was durch das „g“ in der verkürzten Schreibweise der Versionsnummer zum Tragen kommt.[220]

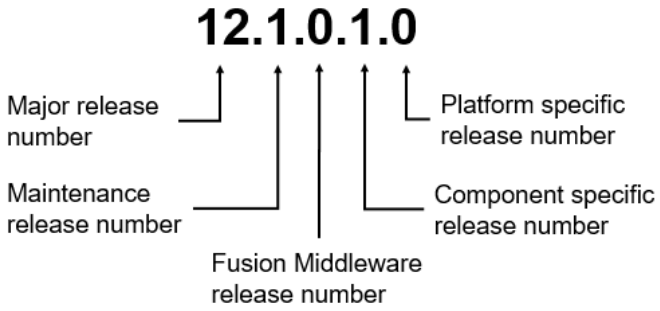

Abbildung 11 - Oracle Database Release Number[221]

Die zweite Zahl der Versionsnummer von links (siehe Abbildung 11) ist die Wartungsversionsnummer (Maintenance Release Number). In der verkürzten Form entsprechend durch den Anhang „Release 1“ (Oracle Database 12c Release 1) dargestellt. Die Wartungsversion enthält Softwareaktualisierungen, die in erster Linie Fehler (Bugs) beseitigen aber auch neuartige Funktionen im Datenbanksystem enthalten können. Die Fusion Middleware release number (dritte Zahl) kennzeichnet Aktualisierungen, die der Oracle Fusion Middleware Plattform zuzuschreiben sind. Diese Middleware umfasst unter anderem den WebLogic Server oder den Oracle Application Server, die wiederum aus weiteren Komponenten bestehen können. Die vierte Zahl von links ist die Component-specific release number, die einen bestimmten Release-Stand (Release Level) der einzelnen Komponenten beschreibt. Eine Erhöhung der Zahl deutet auf neue Software oder Sicherheitsaktualisierungen der Komponenten hin. Die letzte Zahl der Versionsnummer ist die plattformspezifische Veröffentlichungsnummer

[219] Vgl. **[OraAA]**
[220] Vgl. **[OraSR]**
[221] Angelehnt an ebd.

(Platform-Specific Release Number). Diese wird auch als Patch Set Update bezeichnet und enthält kumulierte Priority-, Security- oder auch Bugfixes.[222]

Kritische Aktualisierungen (Critical Patch Updates) und Sicherheitsaktualisierungen (Security Patch Updates) werden bei Oracle im Turnus von 3 Monaten an festgelegten Tagen (z.B. 18. Oktober 2016) dem sogenannten Patchday veröffentlicht (siehe Kapitel 5.1.6). Die Patches werden durch das Oracle Patch Utility (OPatch) nach dem Download (z.B. Patch 17027533 – 12.1.0.1.1 Patch Set Update) manuell eingespielt. Die Prozedur ist nicht benutzerfreundlich, da unter anderem die Reihenfolge der Patches streng eingehalten werden muss. Abhilfe schafft der Enterprise Manager, der OPatch als integralen Bestandteil enthält und das Einspielen von Patchen sehr vereinfacht.[223]

Ein Upgrade auf eine höhere Version (11g auf 12c) wird durch den Database Upgrade Assistant (DBUA) oder durch die manuelle Eingabeaufforderung bewerkstelligt. Das DBUA besitzt eine grafische Oberfläche, die durch die Installationsroutine des Upgrades führt. Der Assistent steht ausschließlich für Direct Upgrades bereit. Beim Direct Upgrade können nur bestimmte gepatchte und upgegradete Versionen, die einer bestimmten Versionsnummer entsprechen (z.B. 10.2.0.5/11.1.0.7/11.2.0.2) auf die neuste Version (12.1.0.1) aktualisiert werden. Dabei besteht die Möglichkeit, das Datenbanksystem durch die ausstehenden Patch Set Updates auf die aktuelle Version (z.B. 10.2.0.5) zu aktualisieren und anschließend das Upgrade durchzuführen. Dazu steht das In-Place Upgrade (Installation in das bestehende ORACLE_HOME Verzeichnis) und das Out-of-Place Upgrade (Installation in ein neues ORACLE_HOME Verzeichnis) zur Verfügung.[224]

Als Indirect Upgrades werden Versionsstände (z.B. 10.2.0.4) bezeichnet, die nicht direkt auf die nächste Version aktualisiert werden können. Hierzu stehen Hilfsmittel wie etwa das Data Pump Utility zur Verfügung, um die gesamten Daten vor der Aktualisierung zu exportieren und danach in eine neue Umgebung zu importieren. Wenn nur ein bestimmter Tabellenraum übernommen werden soll, kann das mit dem Hilfsmittel Transportable Tablespace bewerkstelligt werden. Weiter ermöglicht das Golden Gate Replication Werkzeug die Integration und Replikation von Daten in Echtzeit.[225]

5.4.7 Sicherheitsvorfälle

Die Anzahl der Oracle Database Sicherheitsvorfälle (siehe Tabelle 9) wurden durch eine Auswertung der NVD-Daten der letzten 5 Jahre (Auswertezeitraum 01.

[222] Vgl. **[OraSR]**
[223] Vgl. **[OraPU]**
[224] Vgl. **[Tur15]**
[225] Vgl. ebd.

Januar 2011 bis 31. Dezember 2015) auf Basis des Stichworts „Oracle Database Server“ generiert.

Jahr	Niedrig (0-3)	Mittel (4-6)	Hoch (7-10)	Gesamt
2011	8	37	4	49
2012	2	20	2	24
2013	1	6	11	18
2014	5	29	8	42
2015	3	20	7	30
Gesamt	19	112	32	163

Tabelle 9 - Oracle Sicherheitsvorfälle

In dem ausgewerteten Zeitraum wurden 163 Schwachstellen im Oracle Database Server bekannt. Auffällig ist die große Anzahl von kritischen Sicherheitsverletzungen im ausgewerteten Zeitraum.

Z.B. wurde die kritische Sicherheitslücke CVE-2014-6567 mit dem Schweregrad 9 am 17 Juli 2014 öffentlich bekanntgegeben und erst ein halbes Jahr später im Zuge des „Critical Patch Updates“ am 20. Januar 2015 durch Oracle behoben.[226] Die Schwachstelle war durch einen Speicherüberlauf des DBMS_AW Pakets vorhanden („Unspecified vulnerability in the Core RDBMS component in Oracle Database Server 11.1.0.7, 11.2.0.3, 11.2.0.4, 12.1.0.1, and 12.1.0.2 allows remote authenticated users to affect confidentiality, integrity, and availability via unknown vectors.“).[227]

[226] Vgl. **[Lit15]**
[227] Vgl. **[Nat16]**

6 Bewertung der Datenbanksicherheit

Die Bewertung der Datenbanksicherheit findet auf Grundlage der in Kapitel 4 beschriebenen Sicherheitsmechanismen und der in Kapitel 5 analysierten Datenbanksystemen statt. In die Gewichtung der Bewertung fließen die standardmäßig aktiven Schutzmechanismen und die optionalen Schutzfunktionen der Datenbanksysteme ein. In Tabelle 10 sind die Ergebnisse zusammengefasst; Details werden je Sicherheitsaspekt in den folgenden Unterkapiteln bewertet.

Bewertung der Datenbanksicherheit		MySQL Community Server 5.7.10	MariaDB 10.1.11	PostgreSQL 9.5.1	Oracle 12c Release 1 (12.1) Enterprise
Konfiguration	Standardbenutzer	-	-	-	-
	Standardpasswort	+	+	+	+
	unnötige Erweiterung	+	+	+	+
	sichere Verbindung	-	-	-	-
	Testdatenbank	+	+	-	+
	Logging aktiv	-	-	+	+
	externe Dienste	+	+	+	+
Authentifizierung		PAP (SHA-256) (No-Login) (Socket)	PAP (Socket) (PAM) (PIPE) (SSPI/GSSAPI)	PAP/Socket (Trust) (GSSAPI) (Ident) (Peer) (LDAP) (Radius) (Cert) (Pam)	PAP (Kerberos) (Radius) (SSL)
Zugriffskontrolle	Zugriffskontrolle	DAC	RBAC	RBAC	RBAC
	Objektprivilegien	X	X	DAC	DAC
	Systemprivilegien	X	X	X	CBAC
	Erweiterungen	X	X	(Row-level security) (Sepgsql)	(VPD durch FGAC) (OLS durch MLS) (DV durch MAC) (RAS durch ACL)
Verschlüsselung	Datenbank-verschlüsselung	Column-level encryption	Tablespace encryption Data-at-rest-encryption	Column-level encryption	Column-level encryption Tablespace encryption
	Algorithmus	MD5 SHA-1/SHA-2 AES	MD5 SHA-1/SHA-2 AES DES	MD5 SHA-1/SHA-2 (AES) (3DES) (Blowfish) (CAST5)	MD4/MD5 (SHA-1)/SHA-2 AES 3DES RC4 ODR
	Passwortstärke	SHA-1 (41hex)	SHA-1 (41hex)	MD5	SHA-512 + PBKDF2
	Sicherheitsprotokoll	-	-	-	-
Auditing		Error log (General query log) (Binary log) (Relay log) (Slow query log) (DDL log)	Error log (General query log) (Binary log) (Relay log) (Slow query log) (MyISAM log) (MariaDB Audit Tool)	CONNECT DDL DML Debug ext. A. pg_stat pg_statio	Mandatory Auditing Unified Auditing (Privileged User Auditing) (Standard Auditing) (Fine Grained Auditing) (Database Vault Auditing) (Trigger based Auditing)
Sicherheit	Strategie	Turnus 3 Monate	Direkt & transparent	Direkt & transparent	Turnus 3 Monate
	Sicherheitsvorfälle	281	193	44	163
	- Niedrig	77	43	1	19
	- Mittel	191	142	35	112
	- Hoch	13	8	8	32

- : Sicherheitsrisiko; + : kein Sicherheitsproblem; ( ) : optionale Erweiterung; X : nicht verfügbar

Tabelle 10 - Bewertung der Datenbanksicherheit

6.1 Konfiguration

Die Konfigurationssicherheit wird durch restriktive Konfiguration und Härtung des Datenbanksystems erreicht. Die ersten Schritte nach einer erfolgreichen Installation (defaultmäßige Konfiguration) sollten die manuelle Anpassung und Härtung der Konfigurationsdatei beinhalten. Die Bewertung der Konfiguration beinhaltet die Sicherheit der standardmäßigen Datenbankkonfiguration. Bei keinem der untersuchten Datenbanksysteme war es möglich, während der Konfigurationsroutine den administrativen Benutzernamen (z.B. root, postgres oder SYS) zu ändern. Durch Standardbenutzernamen sind Cyberkriminelle in der Lage, die Passwörter durch simples Ausprobieren zu ermitteln. Der kombinierte Schutz aus unbekanntem Benutzernamen und Passwort ist nach einer Standardinstallation nicht gegeben. Weiter ist bei keinem untersuchten Datenbanksystem die Übertragungssicherheit defaultmäßig aktiviert. MySQL und MariaDB verzichten in der Standardkonfiguration komplett auf das Auditing, wodurch die Nachvollziehbarkeit von Sicherheitsvorfällen nicht möglich ist. Das PostgreSQL-Datenbanksystem stellt standardmäßig eine Testdatenbank bereit, die als Einfallstor zum Einschleusen von beliebigen, auch schädlichen Code (z.B. PHP Shell oder SQL-Injection) genutzt werden kann. Die sicherste Grundkonfiguration stellt das Oracle 12c Datenbanksystem bereit.

6.2 Authentifizierung

Die Authentifizierung schützt das Datenbanksystem vor unautorisierten Zugriffen. Dazu stehen unterschiedliche Authentifizierungsverfahren zur Verfügung. Die Benutzername-Passwort-Authentifizierung (PAP) ist bei allen Datenbanksystemen standardmäßig aktiv. Dieses Verfahren ist das am weitesten verbreitete Authentifizierungsverfahren im World Wide Web. Um die Anmeldeinformationen während der Authentifizierung zu schützen, muss zusätzlich ein Sicherheitsprotokoll (SSL/TLS) für die Datenübertragung verwendet werden. Standardmäßig verfügt keines der untersuchten Datenbanksysteme über eine SSL/TLS-Verschlüsselung. Bei MySQL, MariaDB und PostgreSQL (UNIX) wird somit der Benutzername und das Passwort im Klartext übertragen. Bei PostgreSQL (WINDOWS) wird das Passwort schon clientseitig vor der Übertragung in eine MD5-Hash verschlüsselt. Das Oracle 12c Datenbanksystem verfügt über eine automatische und transparente Passwortverschlüsselung auf Basis des AES-Algorithmus. Starke Authentifizierungsmethoden sind nur in Oracle 12c (Kerberos, Radius) und PostgreSQL (Radius, PKI) vorhanden. Die Zwei-Faktor-Authentifizierung über zwei getrennte Kommunikationskanäle ist bis dato in keinem Datenbanksystem implementiert.

6.3 Zugriffskontrolle

Die Zugriffskontrolle hat die Aufgabe die Benutzerberechtigungen so restriktiv wie möglich zu verwalten. Die rollenbasierte Zugriffskontrollstrategie stellt die benutzerfreundlichste und effektivste Methode der Zugriffskontrolle dar. Bis auf MySQL stellen alle untersuchten Datenbanksysteme die Role-Based Access Control zur Verfügung. Bei MySQL werden die Berechtigungen durch die benutzerbestimmbare Zugriffskontrolle (DAC) verwaltet. In PostgreSQL und Oracle 12c werden Objektprivilegien durch die DAC geregelt. Zusätzlich verfügt Oracle 12c über Systemprivilegien, die mittels der Code-Based Access Control geschützt werden. Als erweiterten Schutz kann bei PostgreSQL die Row-Level Security zur Vergabe von Sicherheitsrichtlinien auf Spaltenebene genutzt werden. Mit Sepgsql wird in PostgreSQL auch noch eine Label Based Mandatory Access Control, die eine weitere Sicherheitsebene speziell für Linux-Betriebssysteme enthält, bereitgestellt. Das Oracle 12c Datenbanksystem verfügt darüber hinaus auch noch über die Virtual Private Database, die der Fine Grained Access Control unterliegt, die Oracle Label Security mittels Multi-Level Security, dem Database Vault, der durch die Mandatory Access Control geschützt wird, und der Real Application Security die der Zugriffssteuerungsliste (ACL) unterliegt.

6.4 Verschlüsselung

Die Datenbankverschlüsselung kann auf unterschiedlichen Datenebenen implementiert werden. Alle untersuchten Datenbanksysteme ausgenommen MariaDB verfügen über die Verschlüsselung auf Spaltenebene (Column-Level Encryption). MariaDB und Oracle 12c verfügen über die Tabellenraum-Verschlüsselung (Tablespace Encryption). Einzig MariaDB ermöglicht die Verschlüsselung der gesamten Datenbank (Data-at-Rest-Encryption) mitsamt den Logdateien. Alle Datenbanksysteme unterstützen zur Verschlüsselung von sensitiven Daten den Advanced Encryption Standard (AES). Die Schlüssellänge stellt immer einen Kompromiss zwischen Ressourcenaufwand (Rechnerleistung) und Sicherheit dar. Der AES-Algorithmus bildet den besten Kompromiss aus Sicherheit und Geschwindigkeit. Bei PostgreSQL muss das Verschlüsselungsmodul vor der ersten Benutzung aktiviert werden. Standardmäßig sind in PostgreSQL nur die Hashalgorithmen MD5 und SHA-1 nutzbar. In der Standardkonfiguration verwendet PostgreSQL den veralteten MD5-Hash-Algorithmus zum Passwortschutz. MySQL und die Abspaltung MariaDB verschlüsseln das Passwort mit dem nicht mehr sicheren SHA-1 Algorithmus. Oracle nutzt zur Passwortsicherheit die kryptographische Hashfunktion SHA-512 und die Password-Based Key Derivation Function 2. Zur Kommunikationssicherheit kann bei allen untersuchten Datenbanksysteme das

TLS V1.2 Verfahren optional eingesetzt werden, da standardmäßig kein Sicherheitsprotokoll aktiv ist.

6.5 Auditing

In der Standardkonfiguration verfügt MySQL und MariaDB über keine aktiven Auditing-Richtlinien. Einzig das Fehlerprotokoll, welches Ereignisse des Datenbankservers protokolliert, ist in MySQL (WINDOWS) und MariaDB aktiviert. MariaDB verfügt zwar über das MariaDB Audit Tool, das aber standardmäßig deaktiviert ist. In der Default-Konfiguration stellt PostgreSQL alle Überwachungs- und Protokollfunktionen standardmäßig bereit. Bei Oracle ist die zwingende Überprüfung (Mandatory Auditing) und das Unified Auditing von Datenbankaktivitäten aktiv. Darüber hinaus verfügen alle Datenbanksysteme über externe Auditing-Erweiterungen, die zusätzlich installiert werden können.

6.6 Sicherheitsupdate

Die Sicherheit von Softwareprodukten spiegelt einen wesentlichen Aspekt der Datenbanksicherheit wieder. Die Softwarehersteller verfolgen dabei unterschiedliche Strategien. MySQL und Oracle Database, die beide zur Oracle Corporation gehören, erhalten Sicherheitsaktualisierungen im Turnus von 3 Monaten an definierten Bereitstellungstagen. Die Softwarefehler werden bei beiden Datenbanksystemen nicht öffentlich bekanntgegeben. MariaDB hält sich eng an den CVE-Standard und legt auch Informationen über die geschlossenen Sicherheitslücken transparent offen. Die Schwachstellen werden je nach Schweregrad zeitnah von den MariaDB-Entwicklern behoben. PostgreSQL verfolgt die gleiche Sicherheitspolitik wie MariaDB und listet die Sicherheitslücken öffentlich per CVE-Standard. Die Softwareaktualisierungen müssen bei allen analysierten Datenbanksystemen manuell eingespielt werden.

6.7 Sicherheitsvorfälle

Die absolut geringste Quote an Sicherheitsverletzungen ist beim PostgreSQL-Datenbanksystem zu finden. Das Datenbanksystem kommt im ausgewerteten fünf Jahreszeitraum (2011-2015) auf 44 bekannt gewordene Sicherheitsvorfälle. Zum Vergleich, MySQL bringt es im gleichen Zeitraum auf mehr als sechsmal so viele Schwachstellen (281 Vorfälle). Oracle enthält viermal so viel kritische Sicherheitslücken (32 Stück) als PostgreSQL oder MariaDB (jeweils nur acht kritische Vorfälle). Hier zeigt sich, dass die Sicherheitsupdatestrategie von PostgreSQL und MariaDB entsprechend positiv wirkt.

Fazit

Der Firmenwert eines Unternehmens wird längst nicht mehr am Kapital oder den Produkten gemessen, vielmehr sind es die Informationen (z.B. Kundendaten) die Wettbewerber unterscheiden.

Für die Informationssicherheit und zur Einhaltung unternehmensinterner, gesetzlicher oder vertraglicher Vorgaben (IT-Compliance) stellen Datenbanksysteme unterschiedliche Datenbanksicherheitslösungen bereit. Die Schutzmechanismen umfassen die Bereiche Authentifizierung, Zugriffskontrolle, Datenverschlüsselung und Auditing. Neben den Schutzvorkehrungen ist jedoch auch eine sorgfältige Installation und Konfiguration sowie permanentes Einspielen von Sicherheitsupdates unabdingbar. Weiter ist es wichtig, ein ganzheitliches Konzept der Datenbanksicherheit zu entwickeln und technisch umzusetzen. Die Schutzvorkehrungen müssen sich nach der Wichtigkeit und Sensibilität der Daten richten. Die Absicherung des Datenbanksystems muss mehrstufig und durch mehrere Mechanismen zur Sicherheit (z.B. Schutz gegen SQL-Injection oder Malware) erfolgen.

Der MySQL Community Server ist ein schlankes und schnelles Open-Source Datenbanksystem, das gerade im Web-Umfeld bevorzugt eingesetzt wird. Durch den einfachen Aufbau und die Struktur eignet es sich, um Projekte schnell und ohne große Hindernisse zu realisieren. Der MySQL Community Server verfügt in der Standardausführung nicht über die Schutzvorkehrungen (z.B. Auditing-Werkzeuge) der Konkurrenzprodukte (vgl. Kap. 6), punktet aber durch eine einfache Handhabung und gute Übersichtlichkeit. Die Abspaltung MariaDB verfügt über die gleichen Eigenschaften, bietet darüber hinaus aber bessere Schutzvorkehrungen, die sich gerade in der rollenbasierten Zugriffskontrolle und der Verschlüsselung auf Tabellen- und Datenbankebene sowie der besseren Update-Politik auszeichnen.

Das PostgreSQL-Datenbanksystem stellt das beste Gesamtpaket der untersuchten Datenbanksysteme bereit (vgl. Kap. 6). Ausschlaggebend für diese Bewertung sind die direkte und transparente Sicherheitspolitik sowie die geringe Anzahl an Sicherheitsverletzungen im ausgewerteten Zeitraum (2011-2015). Das Datenbanksystem verfügt über umfangreiche und sichere Methoden zur Authentifizierung sowie zahlreiche Verschlüsselungsalgorithmen zum Schutz von sensiblen Daten. PostgreSQL hat zwei Schwachpunkte: die schwache Passwortsicherheit (MD5-Hashfunktion) der Standardkonfiguration sowie die nur auf Spaltenebene anwendbare Datenbankverschlüsselung.

Das Oracle Datenbanksystem verfügt über die besten Sicherheitsmethoden bei der Zugriffskontrolle und den Auditing-Werkzeugen. Oracle bietet neben der

rollenbasierten Zugriffssteuerung weitere Sicherheitsmethoden zur Kontrolle der Zugriffe (Virtual Private Database, Label Security, Database Vault und Real Application Security) an. Neben der zwingenden Überprüfung (Mandatory Auditing) der Benutzeraktivitäten verfügt die Oracle Database 12c über das Standard Auditing zur Überwachung von Privilegien und Objekten, dem Fine Grained Auditing zur Kontrolle der Richtlinien auf Spaltenebene sowie das Privileged User Auditing zur Überprüfung administrativer Konten. Einzig die Anzahl der schweren Sicherheitsvorfälle trübt den positiven Gesamteindruck des Datenbanksystems. Im ausgewerteten Zeitraum sind 6,4 kritische Sicherheitslücken pro Jahr im Oracle Datenbanksystem aufgedeckt worden. Viermal mehr als im Vergleich zu MariaDB oder PostgreSQL mit nur je 1,6 kritischen Sicherheitslücken pro Jahr.

Die Ausarbeitung hat gezeigt, dass sich keines der analysierten Datenbanksysteme (MySQL Community Server 5.7.10, MariaDB 10.1.11, PostgreSQL 9.5.1 und Oracle Database 12c) pauschal favorisieren lässt. Die funktionale Sicherheit hängt immer vom Einsatzgebiet, den verwendeten optionalen Sicherheitsfunktionen und den zusätzlich verankerten Schutzmaßnahmen (technischen Faktoren) der Server-Plattform ab. Die größte Gefahr geht von nicht ausreichend gepatchten Datenbanksystemen aus. Der Vergleich und die Bewertung der Datenbanksicherheit relationaler Datenbanksysteme hat gezeigt, dass die Konfiguration/Implementierung des Datenbanksystems wie auch die Softwarequalität des eingesetzten Datenbanksystems einen wesentlichen Faktor für die Sicherheit von sensiblen und personenbezogenen Daten darstellt.

Im Gegensatz zu den untersuchten relationalen Datenbanksystemen verfolgt die objektorientierte Datenbank ZeroDB einen interessanten Alternativansatz, der besonderen Wert auf die Sicherheit und Privatsphäre der Daten legt. Das Datenbanksystem ist unter der freien GNU Affero General Public License (AGPL) erhältlich und verfügt über eine Ende-zu-Ende-Verschlüsselung. Die Besonderheit des Datenbanksystems ist, dass die Datenbanklogik und die Ver- und Entschlüsselung ausschließlich auf dem Client erfolgt.[228] Dadurch werden die Daten standardmäßig verschlüsselt übertragen und verschlüsselt in der Datenbank in B-Bäumen ohne Hinweise auf die hinterlegte Datenstruktur gespeichert. Da der Datenserver nur verschlüsselte und losgelöste, voneinander unabhängige Daten enthält, ist ein Datendiebstahl und Datenmissbrauch im Wesentlichen ausgeschlossen.[229]

[228] Vgl. **[Zero]**
[229] Vgl. **[Men15]**

Literaturverzeichnis

[BIT15] BITKOM: **Kompass der IT-Sicherheitsstandards**, 2015. https://www.bitkom.org/Publikationen/2013/Leitfaden/Kompass-der-IT-Sicherheitsstandards/Kompass-der-IT-Sicherheitsstandards.pdf; Zugriff: 16. März 2016.

[BITKOM] BITKOM: **Digitale Angriffe auf jedes zweite Unternehmen**, 2015. https://www.bitkom.org/Presse/Presseinformation/Digitale-Angriffe-auf-jedes-zweite-Unternehmen.html; Zugriff: 25. April 2016.

[BSI] Bundesamt für Sicherheit in der Informationstechnik: **Sicherheitsmechanismen in elektronischen Ausweisdokumenten**, o.J. https://www.bsi.bund.de/DE/Themen/DigitaleGesellschaft/ElektronischeAusweise/Sicherheitsmechanismen/sicherPKI/pki_node.html; Zugriff: 22. März 2016.

[BSI12] Bundesamt für Sicherheit in der Informationstechnik: **Leitfaden Informationssicherheit**, 2012. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Leitfaden/GS-Leitfaden_pdf.pdf;jsessionid=EBFB27B4CCFAB5F5FDD4216485AD1895.2_cid359?__blob=publicationFile&v=1; Zugriff: 07. April 2016.

[BSI16] Bundesamt für Sicherheit in der Informationstechnik: **IT-Grundschutz-Kataloge**, 2016. https://download.gsb.bund.de/BSI/ITGSK/IT-Grundschutz-Kataloge_2016_EL15_DE.pdf; Zugriff: 18. Mai 2016.

[Bun16] Bundesministerium für Wirtschaft und Energie: **Normen und Standards**, 2016. http://www.bmwi.de/DE/Themen/Technologie/Rahmenbedingungen/normen-und-standards.html; Zugriff: 22. Mai 2016.

[But71] Butler, Lampson W.: **Protection**, 1971. http://research.microsoft.com/en-us/um/people/blampson/08-protection/webpage.html; Zugriff: 28. März 2016.

[Cas95] Castano, S. et al.: **Database Security**, ACM Press, o.O., 1995.

[Clu12] Cluley, Graham: **Millions of LinkedIn passwords reportedly leaked – take action NOW**, 2012. https://nakedsecurity.sophos.com/2012/06/06/millions-of-linkedin-passwords-reportedly-leaked-take-action-now/; Zugriff: 07. Juli 2016.

[Cor11] Cordts, Sänke et al.: **Datenbanken für Wirtschaftsinformatiker**, Vieweg + Teubner Verlag, Wiesbaden, 2011.

[Cor16] Cory, Scott: **Protecting Our Members**, 2016. https://blog.linkedin.com/2016/05/18/protecting-our-members; Zugriff: 07. Juli 2016.

[CVE] The MITRE Corporation: **Common Vulnerabilities and Exposures FAQ**, 2015. https://cve.mitre.org/about/faqs.html#a1; Zugriff: 15. April 2016.

[CVSS] FIRST.Org, Inc.: **Common Vulnerability Scoring System v3.0: Specification Document**, o.J. https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf; Zugriff: 01. Mai 2016.

[Dea15] Dean, Mike: **All About Oracle Auditing - Update for 12c!**, 2015. http://www.dbspecialists.com/files/presentations/OracleAuditing-WhitePaper.pdf; Zugriff: 28. Mai 2016.

[Eis13] Eisentraut, Peter et al.: **PostgreSQL-Administration**, O`Reilly Verlag GmbH & Co. KG, Köln, 2013.

[Els] Elschner, Helmut et al.: **Wie Radius den Netzzugang kontrolliert**, o.J. http://www.heise.de/ix/artikel/Abrechnung-im-Hintergrund-506512.html; Zugriff: 12. April 2016.

[Eng93] Engesser, Hermann et al.: **Duden Informatik: ein Sachlexikon für Studium und Praxis**, Dudenverlag, o.O., 1993.

[Ern15] Ernst, Hartmut et al.: **Grundkurs Informatik**, Springer Vieweg, Wiesbaden, 2015.

[Fab13] Fabry, Heinz-Wilhelm: **Security: Neue Möglichkeiten mit Oracle Database 12c**, 2013. http://www.doag.org/home/aktuelle-news/article/security-neue-moeglichkeiten-mit-oracle-database-12c.html; Zugriff: 23. Mai 2016.

[Far15] Farmer, Todd: **Protecting MySQL Passwords With the sha256_password Plugin**, 2015. http://mysqlserverteam.com/protecting-mysql-passwords-with-the-sha256_password-plugin/; Zugriff: 12. April 2016.

[Fis08] Fischer, Maximilian: **8. Advanced Encryption Standard**, 2008. http://ningelgen.eu/03_Kryptologie/KryptDateien/Kapitel%2008_AES.pdf; Zugriff: 02. April 2016.

[Frö14] Fröhlich, Lutz: **Oracle 12c: Das umfassende Handbuch**, MITP Verlags GmbH, Heidelberg, 2014.

[Gae15] Gaetjen, Scott et al.: **Oracle Database 12c Security**, McGraw-Hill Education, o.O., 2015.

[Ger08] Gertz Michael, Sushil Jajodia: **Handbook of Database Security - Applications and Trends**, Springer Verlag, New York, 2008.

[Gol11] Gollmann, Dieter: **Computer Security**, John Wiley & Sons, Ltd., Hamburg, 2011.

[Gre91] Grebe, R. et al.: **Parallele Datenverarbeitung mit dem Transputer**, Springer-Verlag, Aachen, 1991.

[IBM] IBM: **Kennsatzbasierte Zugriffssteuerung (LBAC)**, o.J. http://www.ibm.com/support/knowledgecenter/de/SSEPGG_9.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0021114.html; Zugriff: 22. März 2016.

[IBM09] IBM: **Verschlüsselung auf Spaltenebene**, 2009. http://www.ibm.com/support/knowledgecenter/de/SSGU8G_11.50.0/com.ibm.sec.doc/ids_ce_001.htm; Zugriff: 05. Mai 2016.

[Jav08] Javie, Francisco et al.: **Advanced Policy-based Access Control in RDBMS**, 2008.
http://www.l3s.de/~zerr/teaching/MasterThesis_Fco_Revilla.pdf; Zugriff: 12. Mai 2016.

[Kea14] Keary, Eoin et al.: **Open Web Application Security Project - Testing Guide**, 2014.
https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf; Zugriff: 22. April 2016.

[Ken15] Kenler, Emilien et al.: **MariaDB Essentials**, Packt Publishing Ltd., Birmingham, 2015.

[Kyt10] Kyte, Thomas: **Expert Oracle Database Architecture: Oracle Database 9i, 10g, and 11g Programming Techniques and Solutions**, Apress, o.O., 2010.

[Lan13] Lane, Adrian: **10 Most Common Security Vulnerabilities In Enterprise Database**, 2013.
http://www.darkreading.com/risk/10-most-common-security-vulnerabilities-in-enterprise-databases/d/d-id/1139979; Zugriff: 15. April 2016.

[Lit15] Litchfield, David: **Oracle Critical Patch Update**, 2015.
http://www.databaseforensics.com/Oracle_Jan2015_CPU.pdf; Zugriff: 24. Juni 2016.

[Mar] MariaDB: **PAM Authentication Plugin**, o.J.
https://mariadb.com/kb/en/mariadb/pam-authentication-plugin/; Zugriff: 12. April 2016.

[Mar16] MariaDB: **named_pipe Authentication Plugin**, 2016.
https://mariadb.com/kb/en/mariadb/named_pipe-authentication-plugin/; Zugriff: 19. April 2016.

[MarAP] MariaDB: **About the MariaDB Audit Plugin**, 2016.
https://mariadb.com/kb/en/mariadb/about-the-mariadb-audit-plugin/; Zugriff: 01. Juni 2016.

[MarASV] MariaDB: **Server_Audit System Variables**, 2016.
https://mariadb.com/kb/en/mariadb/server_audit-system-variables/#server_audit_file_path; Zugriff: 17. April 2016.

[MarDE] MariaDB: **Data at Rest Encryption**, o.J..
https://mariadb.com/kb/en/mariadb/data-at-rest-encryption/; Zugriff: 28. März 2016.

[MarEH] MariaDB: **Encryption, Hashing and Compression Functions**, o.J..
https://mariadb.com/kb/en/mariadb/encryption-hashing-and-compression-functions/; Zugriff: 19. März 2016.

[MarK] MariaDB: **MariaDB versus MySQL – Kompatibilität**, 2016.
https://mariadb.com/kb/de/mariadb-vs-mysql-compatibility/; Zugriff: 24. Juni 2016.

[MarLF] MariaDB: **Log Files**, o.J.
https://mariadb.com/kb/en/mariadb/log-files/; Zugriff: 19. März 2016.

[MarMP] MariaDB: **Maintenance Policy**, 2016.
https://mariadb.org/about/maintenance-policy/; Zugriff: 23. Juni 2016.

[MarOS] MariaDB: **MariaDB Foundation to Safeguard Leading Open Source Database**, 2014.
https://mariadb.org/mariadb-foundation-to-safeguard-leading-open-source-database/; Zugriff: 23. März 2016.

[MarPV] MariaDB: **Password Validation**, o.J.
https://mariadb.com/kb/en/mariadb/password-validation/; Zugriff: 04. April 2016.

[MarR] MariaDB: **Roles Overview**, o.J.
https://mariadb.com/kb/en/mariadb/roles-overview/; Zugriff: 26. März 2016.

[MarRN] MariaDB: **MariaDB 5.5.40 Release Notes**, 2014.
https://mariadb.com/kb/en/mariadb/mariadb-5540-release-notes/; Zugriff: 03. Juni 2016.

[MarSV] MariaDB: **Security Vulnerabilities Fixed in MariaDB**, o.J.
https://mariadb.com/kb/en/mariadb/security/; Zugriff: 03. April 2016.

[MarTE] MariaDB: **Table and tablespace encryption on MariaDB 10.1.3**, o.J..
https://blog.mariadb.org/table-and-tablespace-encryption-on-mariadb-10-1-3/; Zugriff: 29. März 2016.

[MarUS] MariaDB: **UNIX_SOCKET Authentication Plugin**, o.J.
https://mariadb.com/kb/en/mariadb/unix_socket-authentication-plugin/; Zugriff: 30. März 2016.

[Mei12] Meinel, Christoph et al.: **Internetworking: Technische Grundlagen und Anwendungen**, Springer-Verlag, Heidelberg, 2012.

[Mel07] Mell, Peter et al.: **A Complete Guide to the Common Vulnerability Scoring System Version 2.0**, 2007.
https://www.first.org/cvss/cvss-v2-guide.pdf; Zugriff: 06. Mai 2016.

[Men15] Menge-Sonnentag, Rainald: **Verschlüsselnde Datenbank ZeroDB wird Open Source**, 2015.
http://www.heise.de/developer/meldung/Verschluesselnde-Datenbank-ZeroDB-wird-Open-Source-3034728.html; Zugriff: 02. Juni 2016.

[Mey13] Meyers, Steve: **MySQL Forks: A Brief History of MySQL**, 2013.
http://www.stevemeyers.net/2013/11/a-brief-history-of-mysql.html; Zugriff: 17. März 2016.

[Mic11] Microsoft TechNet: **Bewertungssystem für Security Bulletins des Microsoft Security Response Center**, 2011.
https://technet.microsoft.com/de-de/security/gg309177.aspx; Zugriff: 15. Juni 2016.

[Mil14] Miller, Michae A. et al.: **White Paper - Oracle 12c Unified Auditing**, 2016.
http://www.integrigy.com/files/Integrigy%20Oracle%2012c%20Unified%20Auditing.pdf; Zugriff: 26. Mai 2016.

[Mor10] Morgan, Andrew G. et al.: **The Linux-PAM System Administrators Guide**, 2010.
http://www.linux-pam.org/documentation/Linux-PAM-1.2.0-docs.tar.gz; Zugriff: 15. Mai 2016.

[Müt13] Mützlitz, Carsten: **Oracle Security in der Praxis: Vollständige Sicherheitsüberprüfung für Ihre Oracle-Datenbank**, Carl Hanser Verlag, Falkensee, 2013.

[MyS15] MySQL™: **MySQL 5.7 Reference Manual**, 2015. http://downloads.mysql.com/docs/refman-5.7-en.pdf; Zugriff: 02. April 2016.

[Nat10] Natan, Ben Ron: **Acht Schritte zur ganzheitlichen Daenbanksicherheit**, 2010. ftp://public.dhe.ibm.com/software/emea/de/db2/IBM_Guardium_Whitepaper_DE.pdf; Zugriff: 07. Juni 2016.

[Nat16] National Institute of Standards and Technology: **Vulnerability Summary for CVE-2014-6567**, 2016. https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-6567; Zugriff: 02. Juni 2016.

[Neu94] Neumann, Clifford B. et al.: **Kerberos: An Authentication Service for Computer Networks**, 1994. http://gost.isi.edu/publications/kerberos-neuman-tso.html; Zugriff: 11. April 2016.

[NVD] National Institute of Standards and Technology: **National Vulnerability Database**, o.J. https://nvd.nist.gov/home.cfm; Zugriff: 25. Mai 2016.

[Oeh15] Oehrli, Stefan: **Oracle 12c Security Features - Datenbank Security im Überblick**, 2015. http://www.trivadis.com/sites/default/files/downloads/soug_sig_oracle_12c_security_features.pdf; Zugriff: 05. Mai 2016.

[Opp13] Oppliger, Rolf: **Computersicherheit: Eine Einführung**, Vieweg, Braunschweig/Wiesbaden, 2013.

[Ora] Oracle Corporation: **Check and Upgrade MySQL Tables**, o.J. http://dev.mysql.com/doc/refman/5.7/en/mysql-upgrade.html; Zugriff: 24. April 2016.

[Ora07] Oracle: **Defying Conventional Wisdom**, 2007. http://www.oracle.com/us/corporate/profit/p27anniv-timeline-151918.pdf; Zugriff: 27. April 2016.

[Ora14] Oracle Corporation: **Changes in MySQL 5.7.4 (2014-03-31, Milestone 14)**, 2014. https://dev.mysql.com/doc/relnotes/mysql/5.7/en/news-5-7-4.html#mysqld-5-7-4-security; Zugriff: 14. Mai 2016.

[OraAA] Oracle: **Database Security Guide - Configuring and Administering Auditing**, 2016. https://docs.oracle.com/cd/B19306_01/network.102/b14266/cfgaudit.htm#BABCFIHB; Zugriff: 31. Mai 2016.

[OraAM] Oracle: **Database Advanced Security Administrator's Guide - Configuring Multiple Authentication Methods and Disabling Oracle Advanced Security**, 2016. https://docs.oracle.com/cd/B19306_01/network.102/b14268/asoauth.htm#g1008047; Zugriff: 15. Mai 2016.

[OraAP] Oracle: **Database Security Guide - Configuring Audit Policies**, 2016. https://docs.oracle.com/database/121/DBSEG/audit_config.htm#DBSEG356; Zugriff: 31. Mai 2016.

[OraAS] Oracle: **Encryption and Redaction in Oracle Database 12c with Oracle Advanced Security**, 2013. http://www.oracle.com/technetwork/database/options/advanced-security/advanced-security-wp-12c-1896139.pdf; Zugriff: 06. Mai 2016.

[OraAT] Oracle: **Database Security Guide - Administering the Audit Trail**, 2016. https://docs.oracle.com/database/121/DBSEG/audit_admin.htm#DBSEG361; Zugriff: 30. Mai 2016.

[OraC] Oracle: **Database PL/SQL Packages and Types Reference - DBMS_CRYPTO**, 2016. https://docs.oracle.com/database/121/ARPLS/d_crypto.htm#ARPLS65670; Zugriff: 23. Juni 2016.

[OraCA] Oracle: **Database Security Guide - Configuring Authentication**, 2016. https://docs.oracle.com/database/121/DBSEG/authentication.htm#DBSEG33223; Zugriff: 17. Mai 2016.

[OraCP] Oracle: **Critical Patch Updates, Security Alerts and Third Party Bulletin**, o.J. http://www.oracle.com/technetwork/topics/security/alerts-086861.html; Zugriff: 17. Mai 2016.

[OraDI] Oracle: **Database Installation Guide - Overview of Oracle Database Installation**, 2016. https://docs.oracle.com/database/121/LADBI/install_overview.htm#LADBI7444; Zugriff: 01. Mai 2016.

[OraDV] Oracle: **Oracle Database Vault with Oracle Database 12c**, 2015. http://www.oracle.com/technetwork/database/options/database-vault/database-vault-wp-12c-1896142.pdf?ssSourceSiteId=ocomde; Zugriff: 12. Mai 2016.

[OraFGA] Oracle: **Database PL/SQL Packages and Types Reference - DBMS_FGA**, 2016. http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/d_fga.htm#CDEIECAG; Zugriff: 28. Mai 2016.

[OraIA] Oralce: **Database Security Guide - Introduction to Auditing**, 2016. https://docs.oracle.com/database/121/DBSEG/auditing.htm#DBSEG1023; Zugriff: 27. Mai 2016.

[OraIR] Oracle: **Database Security Guide- Managing Security for Definer's Rights and Invoker's Rights**, 2016. http://docs.oracle.com/database/121/DBSEG/dr_ir.htm#DBSEG660; Zugriff: 16. Mai 2016.

[OraLS] Oracle: **Oracle Label Security Architecture**, 2015. https://docs.oracle.com/database/121/OLSAG/intro.htm#OLSAG043; Zugriff: 16. Mai 2016.

[OraOLS] Oracle: **Label Security Administrator's Guide - Introduction to Oracle Label Security**, 2016. https://docs.oracle.com/database/121/OLSAG/intro.htm#OLSAG001; Zugriff: 23. Mai 2016.

[OraPL] Oracle: **Oracle Technology Global Price List**, 2015. http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf; Zugriff: 30. April 2016.

[OraPU] Oracle: **Mehr als nur Patching - Das Oracle Patch Utility**, 2011. http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/opatch/index.html; Zugriff: 05. April 2016.

[OraRAS] Oracle: **Database Real Application Security Administrator's and Developer's Guide - Introducing Oracle Database Real Application Security**, 2016. https://docs.oracle.com/database/121/DBFSG/intro.htm#DBFSG99011; Zugriff: 29. Mai 2016.

[OraSA] Oracle: **Database Security Guide - Introduction to Strong Authentication**, 2016. https://docs.oracle.com/database/121/DBSEG/asotools.htm#DBSEG451; Zugriff: 28. Mai 2016.

[OraSD] Oracle: **Oracle Database Software Downloads**, o.J. http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html; Zugriff: 03. Mai 2016.

[OraSR] Oracle: **Oracle® Database Administrator's Guide - Identifying Your Oracle Database Software Release**, 2010. https://docs.oracle.com/cd/E18283_01/server.112/e17120/dba004.htm; Zugriff: 26. Mai 2016.

[OraSSL] Oracle: **Database Security Guide - Configuring Secure Sockets Layer Authentication**, 2016. https://docs.oracle.com/database/121/DBSEG/asossl.htm#DBSEG070; Zugriff: 12. Mai 2016.

[OraVPD] ORACLE Deutschland GmbH: **Oracle Virtual Private Database**, o.J. http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/vpd/index.html; Zugriff: 01. Juni 2016.

[OWASP] The Open Web Application Security Project: **OWASP Top 10-2013**, 2013. https://www.owasp.org/images/4/42/OWASP_Top_10_2013_DE_Version_1_0.pdf; Zugriff: 26. März 2016.

[Pet] Petritsch, Helmut: **Aktuelle Angriffe auf SHA und MD5**, o.J. http://petritsch.co.at/download/Attacken_auf_MD5uSHA1.pdf; Zugriff: 06. April 2016.

[Poh] Pohl, Hartmut: **Less Than Zero Day Vulnerabilities**, o.J. http://inf-red.h-brs.de/informatikmedia/Downloads/Personen/pohl/Aufsaetze/Less_Than_Zero_Day_Vulnerabilities.pdf; Zugriff: 24. März 2016.

[Pon16] Ponemon Institute: **2016 Ponemon Cost of Data Breach Study**, 2016. http://www-03.ibm.com/security/data-breach/; Zugriff: 01. Juli 2016.

[Pos15] The PostgreSQL Global Development Group: **PostgreSQL 9.4.7 Documentation**, 2016. https://www.postgresql.org/files/documentation/pdf/9.4/postgresql-9.4-A4.pdf; Zugriff: 04. Mai 2016.

[PosAW] The PostgreSQL Global Development Group: **PostgreSQL, Award Winning Software**, 2016.
http://www.postgresql.org/about/awards/; Zugriff: 22. April 2016.

[PosD] The PostgreSQL Global Development Group: **PostgreSQL Core Distribution**, 2016.
http://www.postgresql.org/download/; Zugriff: 22. April 2016.

[PosL] The PostgreSQL Global Development Group: **PostgreSQL - License**, 2016.
http://www.postgresql.org/about/licence/; Zugriff: 21. April 2016.

[PosPG] The PostgreSQL Global Development Group: **PostgreSQL - pg_upgrade**, 2016.
http://www.postgresql.org/docs/current/static/pgupgrade.html; Zugriff: 20. April 2016.

[PosRSP] The PostgreSQL Global Development Group: **Row Security Policies**, 2016.
http://www.postgresql.org/docs/current/static/ddl-rowsecurity.html; Zugriff: 24. Mai 2016.

[PosSI] The PostgreSQL Global Development Group: **PostgreSQL - Security Information**, 2016.
http://www.postgresql.org/support/security/; Zugriff: 20. April 2016.

[PosVP] The PostgreSQL Global Development Group: **PostgreSQL - Versioning policy**, 2016.
http://www.postgresql.org/support/versioning/; Zugriff: 21. April 2016.

[Raz14] Razzoli, Federico: **Mastering MariaDB**, Packt Publishing Ltd, Birmingham, 2014.

[RFC1321] Rivest, R.: **The MD5 Message-Digest Algorithm**, 1992.
http://tools.ietf.org/html/rfc1321?ref=driverlayer.com; Zugriff: 04. Mai 2016.

[RFC1334] Lloyd, B. et al.: **PPP Authentication Protocols**, 1994.
https://tools.ietf.org/html/rfc1334; Zugriff: 18. Mai 2016.

[RFC1510] Kohl, J. et al.: **The Kerberos Network Authentication Service (V5)**, 1993.
https://tools.ietf.org/html/rfc1510; Zugriff: 30. April 2016.

[RFC2144] Adams, C.: **The CAST-128 Encryption Algorithm**, 1997.
https://tools.ietf.org/html/rfc2144; Zugriff: 27. April 2016.

[RFC2459] Housley, R. et al.: **Internet X.509 Public Key Infrastructure**, 1999.
http://tools.ietf.org/html/rfc2459; Zugriff: 27. April 2016.

[RFC2865] Rigney, C. et al.: **Remote Authentication Dial In User Service (RADIUS)**, 2000.
https://tools.ietf.org/html/rfc2865; Zugriff: 05. Mai 2016.

[RFC4422] Melnikov, A. et al.: **Simple Authentication and Security Layer (SASL)**, 2006.
https://tools.ietf.org/html/rfc4422; Zugriff: 29. April 2016.

[RFC4510] Zeilenga, K., et al.: **Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map**, 2006.
https://tools.ietf.org/html/rfc4510; Zugriff: 05. Mai 2016.

[RFC4513] Harrison, R. et al.: **Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms**, 2006.
https://tools.ietf.org/html/rfc4513; Zugriff: 13. April 2016.

[RFC5021] Josefsson, S.: **Extended Kerberos Version 5 Key Distribution Center (KDC) Exchanges over TCP**, 2007.
https://tools.ietf.org/html/rfc5021; Zugriff: 26. April 2016.

[RFC6101] Freier, A. et al.: **The Secure Sockets Layer (SSL) Protocol Version 3.0**, 2011.
https://tools.ietf.org/html/rfc6101; Zugriff: 19. Mai 2016.

[Ros08] Ross, Anderson J.: **Security Engineering – A Guide to Building Dependable Distribute**, John Wiley & Sons, o.O., 2008.

[Röc07] Röcher, Dror-John: **Metrikbasiertes Patchen mit CVSS 2.0**, 2007.
https://www.ernw.de/wp-content/uploads/ERNW_Newsletter_19_CVSS_de.pdf; Zugriff: 26. April 2016.

[SAP15] : **SAP Angriffsvektoren - Onapsis Studie**, 2015.
https://www.info-point-security.com/sites/default/files/onapsis-studie_sap-angriffsvektoren.pdf; Zugriff: 18. März 2016.

[Sch02] Schwenk, Jörg: **Sicherheit und Kryptographie im Internet: Von sicherer E-Mail bis zu IP**, Springer Vieweg, Wiesbaden, 2002.

[Sch05] Schneier, Bruce: **Description of a new variable-length key, 64-bit block cipher (Blowfish)**, 2005.
http://link.springer.com/chapter/10.1007%2F3-540-58108-1_24; Zugriff: 15. April 2016.

[Sch06] Schmitz, Roland et al.: **Kompendium Medieninformatik: Mediennetze**, Springer Verlag, Heidelberg, 2006.

[Sch1] Schmidt, Jürgen: **LinkedIn-Passwort-Leck hat desaströse Ausmaße**, 2016.
http://www.heise.de/security/meldung/LinkedIn-Passwort-Leck-hat-desastroese-Ausmasse-3210793.html; Zugriff: 07. Juli 2016.

[Sch10] Schatten, Alexander et al.: **Best Practice Software-Engineering: Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen**, Spektrum Akademischer Verlag, Heidelberg, 2010.

[Sch13] Schmitz, Ludger: **Die Zukunft von MySQL gestalten wir**, 2013.
http://www.computerwoche.de/a/die-zukunft-von-mysql-gestalten-wir,2533344,2#; Zugriff: 07. April 2016.

[Sch16] Scherf, Thorsten: **Gut bewacht**, 2016.
http://www.heise.de/ix/artikel/Gut-bewacht-506652.html; Zugriff: 29. Juni 2016.

[SchJ] Schmidt, Jürgen: **LinkedIn-Leck: Mehr als 80 Prozent der Passwörter bereits geknackt**, 2016.

http://www.heise.de/security/meldung/LinkedIn-Leck-Mehr-als-80-Prozent-der-Passwoerter-bereits-geknackt-3212075.html; Zugriff: 07. Juli 2016.

[Sec14] SecurEnvoy: **What is 2FA?**, 2014. https://www.securenvoy.com/two-factor-authentication/what-is-2fa.shtm; Zugriff: 04. April 2016.

[SEP] : **Security Enhanced PostgreSQL**, o.J. https://code.google.com/p/sepgsql/; Zugriff: 14. Mai 2016.

[Sol08] Sollbach, Wolfgang et al.: **Information Lifecycle Management: Prozessimplementierung**, Springer Verlag, Heidelberg, 2008.

[Spe08] Spenneberg, Ralf: **SELinux & AppArmor: Mandatory Access Control für Linux einsetzen und verwalten**, Addison-Wesley, München, 2008.

[Spi11] Spitz, Stephan et al.: **Kryptographie und IT-Sicherheit: Grundlagen und Anwendungen**, Springer Vieweg, Wiesbaden, 2011.

[Sym16] Symantec: **Internet Security Threat Report**, 2016. https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf; Zugriff: 01. Juni 2016.

[Tho11] Thomas, Tom et al.: **Network Security First-Step**, Cisco Press, Indianapolis, 2012.

[Tie13] Tiemeyer, Ernst: **Handbuch IT-Management: Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis**, Carl Hanser Verlag GmbH Co KG, München, 2013.

[Tip05] Tipton, Harold F. et al.: **Information Security Management Handbook**, Auerbach Publications, o.O., 2005.

[Tur15] Turakhiya, Mitesh: **Database Upgrade**, 2015. http://oraclesupportprovider.blogspot.de/2015/05/database-upgrade.html; Zugriff: 06. Juni 2016.

[Zero] ZeroDB inc.: **ZeroDB**, 2015. https://docs.zerodb.io/; Zugriff: 04. Juni 2016.

Eidesstattliche Erklärung

„Ich versichere an Eides Statt, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen“.

_____ _____

(Ort/Datum) (Unterschrift)